

# **ORACLE RMAN (RECOVERY MANAGER)**

## Contents

1. RMAN NEDİR?.....	3
1.1. RMAN'ın Avantajları: .....	3
1.2. RMAN'ın Dezavantajları: .....	3
1.3. RMAN Neler Yapabiliyor? .....	3
2. RMAN İLE YEDEK ALMAK .....	4
2.1. RMAN ile Cold Backup Şeklinde Yedek Almak.....	4
2.2. Full Database Online Backup .....	4
2.3. Tablespace ve Datafile Yedeklemek .....	5
2.4. Arşiv Dosyalarını Yedeklemek .....	5
3. İLERİ DÜZEY YEDEK İŞLEMLERİ.....	8
3.1. Alınan Yedeklere Etiket Vermek .....	8
3.2. Sıkıştırıp Yedek Almak .....	8
3.3. Alınan Yedeğe Arşiv Dosyalarını Ekleme.....	8
3.4. En Gelişmiş Yedek Alma Komutu .....	8
3.5. Alınan Yedeklerin Hâlâ Kullanılabilir Olup Olmadığını Kontrol Etmek.....	9
3.6. Backup Piece Mantığı .....	9
3.7. Backup Piece'lerin Lokasyonunu Değiştirme.....	9
3.8. Konfigürasyon Ayalarını Değiştirme .....	10
4. YEDEK BİLGİLERİNİ GÖRME .....	11
5. YEDEKTEN DÖNME İŞLEMLERİ.....	12
5.1. Yedek Dosyalarının Doğruluğunu Kontrol Etme .....	12
5.2. Full Recover (Tam Yedekten Dönüş): .....	12
5.3. Datafile'ların ve Tablespace'lerin Recover Edilmesi.....	12
5.4. Arşiv Dosyalarını Yedekten Dönme .....	13
5.5. CONTROL FILE'i Yedekten Dönme .....	13
5.6. Until Time Şeklinde Yedekten Dönme .....	14
5.7. REDO LOG'ların Olmamasının Sorun Yaratması Durumunda .....	14
6. CATALOG MANTIĞI .....	15
7. DEĞERLENDİRME.....	17

## 1. RMAN NEDİR?

RMAN (Recovery Manager), Oracle'in sunduğu bir yedek alma aracıdır. RMAN'ı diğer yöntemlere göre farklı kılan; DBA üzerindeki iş yükünü hafifletmesidir.

RMAN yedek almayla ilgili birçok işi otomatik yapabilme imkanı sunar. RMAN olmaksızın, yedek alma işleminde script'lerin yazılması, tablespace'lerin backup mode'a çekilmesi vb. birçok durumla uğraşmak zorunda kalınabilir. Hâlbuki Oracle'in sunduğu bu araçla yedek alınması için yapılan bütün hazırlık işlemleri otomatik gerçekleşir. Oracle'da yedek alma işlemini düz vites araba kullanmaya benzetirsek, RMAN otomatik vites bir araç gibi düşünülebilir. Vites geçişlerini kendisi ayarlar, sizin bütün yapmanız gerek pedala basmaktır.

### 1.1. RMAN'in Avantajları:

- Backup alma işlemlerini daha basite indirger. Örneğin hotbackup almak için tablespace'leri backup mode'a çekeyim; sonra tekrar normal mode'a alırım gibi dertleriniz olmaz.
- RMAN hotbackup şeklinde veritabanı açıkken alınan yedeklerde ekstra arşiv üretmez. (Gerçekten ekstra arşiv üretmiyor ama yüklü işlemler söz konusu ise hız düşüşü yaşıyor.)
- RMAN ile merkezi tek bir noktadan düzinelerce veritabanına bağlanabilir; her birini yedekleyebilirsiniz. Merkezi kontrol söz konusudur; tek tek her makinede script yazmanız gerekmez. Bir sunucu belirler, her şeyi oradan yapabilirsiniz.
- Kendiliğinden sıkıştırma özelliği var. Yedek alırken, aldığınız yedeği sıkıştırabiliyorsunuz.

### 1.2. RMAN'in Dezavantajları:

- Arka planda tam olarak neler yaptığı açık değil. Anlamak için kurcalamak gerekiyor.
- Hemen hemen her isteğinizi karşılarsa da, belirlenen kalıpların dışına çıkamıyorsunuz. Örneğin bir hotbackup script'i yazdığınızı düşünelim. Hotbackup alınan script için her türlü parametreyi istediğiniz gibi belirleyebilirsiniz. Ancak RMAN ile bu mümkün değil.
- RMAN ile alınan yedekleri, mutlaka RMAN ile recover etmeniz gerekiyor.

### 1.3. RMAN Neler Yapabiliyor?

Söz konusu yedek almak ve yedekten dönmek ise, RMAN hemen hemen her isteğinizi yerine getirebiliyor. Örneğin bozulmuş bir tablespace'i ya da datafile'i veritabanını kapatmaya dahi gerek duymaksızın recover edebiliyorsunuz; ya da veritabanını 3 adımda recover etme imkanına sahipsiniz. Yedek alma esnasında, yedeği Hotbackup şeklinde ya da coldbackup şeklinde de alabiliyorsunuz. Controlfile'ları yedekleyip, daha sonra kullanabiliyorsunuz. Manuel olarak yapılan her backup işleminin, RMAN'de bir karşılığı mevcut.

## 2. RMAN İLE YEDEK ALMAK

RMAN, aldığı yedekleri sadece kendisinin anladığı bir formatta tutuyor. RMAN'ın oluşturduğu bir yedek setini (backupset) unzip edip dilediğiniz gibi kullanma şansına sahip değilsiniz. RMAN ile alınan bir yedek varsa; yedekten dönüş kısmı yine RMAN ile yapılmak zorunda.

RMAN'ın bir veritabanına bağlanması için aşağıdaki komutu, shell üzerinden girmek gerekiyor:

```
$ rman target /
```

Burada önemli olan nokta, hangi işletim sistemi kullanıcısı ile bu komutu çalıştırıyorsanız, o kullanıcıya ait veritabanıyla ilgili çalışacağınızdır. Eğer doğru işletim sistemi kullanıcısı ile komut girilmezse, yanlış bir veritabanı yedeği alırsınız. Şayet yanlış veritabanıyla ilgili çalışma yapmadığınızdan emin olmak isterseniz, TNS adını da kullanabilirsiniz:

```
$ rman target sys/password@YDKTST;
```

**YDKTST**, tnsnames.ora dosyasında tanımlıdır ve yedek alacağımız veritabanını temsil eder.

### 2.1. RMAN ile Cold Backup Şeklinde Yedek Almak

Standart bir cold backup, veritabanını kapatıp, dosyaları bit-bit kopyalamaktan ibarettir. RMAN'de ise veritabanını program içinden kapatıp, backup komutunu vermeniz gerekir. Aşağıdaki shell script ile RMAN üzerinden cold backup alma yöntemini görebilirsiniz:

```
#!/bin/sh
# RMAN COLD BACKUP
rman target / << EOF
shutdown immediate
startup mount;
backup database format '/backup/path/%d_%t_%s.rman';
alter database open;
EOF
exit
```

Yukarıdaki script ile veritabanının full backup'ini alabiliyorsunuz. Farkedeceğimiz gibi yedek alma işlemi sırasında veritabanı kullanılmıyor. Kullanımı basit bir yöntem ancak çalıştığımız ortam gereği, cold backup pek tercih edeceğimiz bir yapı değil. Genellikle online backup şeklinde, veritabanını kapatmadan yedek almamız lâzım.

### 2.2. Full Database Online Backup

RMAN ile bağlantı kurduktan sonra, aşağıdaki komutu girerek database'in tam bir yedeğini almanız mümkündür:

```
RMAN> backup database;
```

Yukarıda gördüğümüz gibi tek satırda veritabanını çalışır vaziyette backup'layabiliyoruz. Ancak nereye, ne şekilde yedekleneceğini belirtmedik. Bu durumda \$ORACLE\_HOME/dbs altında yedek dosyaları oluşacaktır. Çoğu zaman bu hoş olmayan sonuçlar doğurabilir. Disk dolabilir ya da format biçimi istediğiniz gibi olmaz vs... Bu nedenle format parametresini belirtmek her zaman için daha iyidir.

```
RMAN> backup database format '/backup/path/%d_%t_%s.rman';
```

Yedekleme komutunu yukarıdaki gibi girerseniz RMAN yedekleri /backup/path/ altına atılacaktır. %d, %t, %s gibi format biçimleri RMAN'in format yapısıyla alakalıdır. %s yerine %U da kullanılabilir. Ya da sadece %U kullanılabilir. Önemli olan ismin tekrar etmesini engelleyecek şekilde format biçimi girmektir. Format biçimlerini [formatSpec](#) bağlantısında bulabilirsiniz.

### 2.3. Tablespace ve Datafile Yedeklemek

Örnek üzerinden anlatmak daha uygun olacak. System ve Users isimli tablespace'leri yedeklemek istediğimizi düşünelim. Bunun için aşağıdaki işlemi yapmak yeterli:

```
RMAN> backup tablespace system, users;
```

Bu komutu girdiğiniz takdirde, rman belirtilen tablespace'lerin yedeğini alacaktır. Hangi datafile'lar, hangi tablespace'lere bağlı ya da tablespace backup mode'da mı diye bakmanız gerekmez. Bunu rman alt yapısında hâlledecektir. Elbette format belirterek, bir klasör belirtilmesi daha uygun olur.

Şimdi de diğer bir konu, datafile yedeklerinin alınmasına değinelim. Önce hangi datafile'ların yedek alınması gerektiğine karar vermemiz gerekiyor. Bunun için aşağıdaki sorguyu sql ortamında çalıştırabilirsiniz:

```
SQL> SELECT tablespace_name, file#, name  
FROM v$datafile_header ORDER BY tablespace_name;
```

Burada tablespace'leri göstermemizin özel bir nedeni yok. Ben daha düzenli bir yapıda görmek istediğim için bu şekilde bir sorgu kullanıyorum. Hâlbuki, yedek almak istediğiniz datafile'in dosya numarası yeterli. Dilerseniz tablespace'e ait bütün datafile'ları ya da sadece birkaç farklı datafile'i yedekleyebilirsiniz.

Sorgu sonucunda dönen değerlere göre, datafile numaralarını alıp yedekleme işlemine başlıyoruz:

```
RMAN> backup datafile 1,3,45;
```

Komutu yukarıdaki gibi çalıştırırsanız, 1, 3 ve 45 numaralı dosyaları backup'layacaktır. Format belirtmek yine opsiyonel olmakla birlikte, belirtilmesi daha uygundur.

### 2.4. Arşiv Dosyalarını Yedeklemek

Arşiv dosyalarını yedeklemek çok gerekli bir durum olarak görünmeyebilir. Çünkü arşiv dosyaları zaten yedek alma işlemi için kullanılan bir dosya türü. Ancak daha derli toplu olmasını, arşiv dosyalarını tarihe göre kontrol etmeyi ya da daha az yer kaplamasını isteyebilirsiniz. Bu sebeplerle arşiv dosyalarını yedeklemek uygun bir seçenek olabilir. Bunun için aşağıdaki gibi yedekleme yapabilirsiniz:

```
RMAN> BACKUP ARCHIVELOG ALL;
```

Yukarıdaki komut bütün arşiv dosyalarını yedekleyecektir. Fakat arşiv dosyalarının topyekün yedeklenmesine ihtiyaç duymuyor, sadece belirli bir güne kadar olan arşiv dosyalarını tutmak istiyorsanız, aşağıdaki gibi yedekleyebilirsiniz:

```
RMAN> BACKUP ARCHIVELOG FROM TIME='SYSDATE-7'
```

```
FORMAT '/backup/arc_%d_%t_%s.rman';
```

Son girilen ifade, arşiv dosyaları /backup/ altında yedekler. Ancak bütün arşiv dosyaları yedeklenmez; sadece son 7 günlük döneme ait arşiv dosyaları saklanır. Bunun anlamı 1 hafta içinde dilediğiniz bir zamana arşiv dosyalarını kullanarak recover edebileceğinizdir.

Komutu daha farklı da kullanabiliriz; dilersek SCN ya da Sequence Number ile yedekleme yapabilir, ya da 1 günden eski 3 günden yeni arşiv dosyalarını yedekleyebiliriz. Farklı kullanımları gösteren bazı örnekleri aşağıda bulabilirsiniz:

- **SCN'e göre yedekleme yapmak:**

```
RMAN> BACKUP ARCHIVELOG UNTIL SCN = 320
RMAN> BACKUP ARCHIVELOG SCN BETWEEN 205 AND 320
```

- **Arşiv Dizi Numarasına göre yedekleme yapmak:**

```
RMAN> BACKUP ARCHIVELOG UNTIL SEQUENCE = 501;
RMAN> BACKUP ARCHIVELOG FROM SEQUENCE 591
```

- **Zamana göre yedekleme yapmak:**

```
RMAN> BACKUP ARCHIVELOG UNTIL TIME 'SYSDATE-7';
RMAN> BACKUP ARCHIVELOG
FROM TIME 'SYSDATE-30' UNTIL TIME 'SYSDATE-7';
```

- **Yedekledikten sonra, yedeklenen arşiv dosyalarını silmek:**

```
RMAN> BACKUP ARCHIVELOG ALL DELETE INPUT;
```

Özellikle son verilen örnek oldukça önemli. Çünkü son komutla birlikte arşiv dosyalarını yedekledikten sonra silme imkanını kullanıyoruz. Kullanıcı önsezisine gerek olmadan, yedeği alınan dosyanın gereksiz yere diskte bulunmasının önüne bu sayede geçebiliyoruz. Diğer ifadelerde DELETE INPUT koyduğunuz takdirde yedeklenen dosyaların silindiğini göreceksiniz.

## 2.5. Control File Yedeklemek

RMAN ile backup alırken, bazı dosyaların (örneğin controlfile ya da spfile gibi) otomatik olarak yedek alınmasını sağlayabilirsiniz. Aşağıdaki gibi bir giriş yaptığınız takdirde RMAN ile alınan her yedek esnasında, control file da yedeklenecektir.

```
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP ON;
```

Yalnız yedeklendiği klasör \$ORACLE\_HOME/dbs şeklinde olacaktır. Yedeklerinizi doğal olarak farklı bir ortamda tutmak isteyebilirsiniz. Farklı bir konum için aşağıdaki gibi belirtmeniz gerekir:

```
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO
'/data2/backup/cf_%F.rman';
```

Dilerseniz manuel anlık bir backup almanız da mümkündür:

```
RMAN> BACKUP CURRENT CONTROLFILE;
```

Bunu sıklıkla yapmak gerekmiyor. Aldığınız yedeklerin içinde otomatik alındığına dikkat etmeniz yeterli.

### 3. İLERİ DÜZEY YEDEK İŞLEMLERİ

Bu bölümde daha ileri düzey yedek alma işlemlerini konu edeceğiz.

#### 3.1. Alınan Yedeklere Etiket Vermek

Yedek alındığı zaman neyin ne yedeği olduğunu karıştırmamak için RMAN'de TAG yani '*etiket*' isminde bir parametre bulunmaktadır. Örneğin aldığımız full database backup'a bir TAG/etiket ataması yapalım:

```
RMAN> BACKUP DATABASE format '/data2/backup/dbf_%d_%t_%s.rman'
TAG='DBF_CCEBI_20080620_1345'
```

Görüldüğü gibi daha önceki yapıdan hiçbir farkı yok; sadece TAG isminde ekstra bir parametre belirtiliyor.

#### 3.2. Sıkıştırıp Yedek Almak

RMAN ile ilgili en beğendiğim özelliklerden bir tanesi, yedek alırken, dosyaları sıkıştırabilmesidir. Ekstra bir aşama gerektirmez, sadece girilen komuta yeni bazı parametreler gelir:

```
RMAN> BACKUP AS COMPRESSED BACKUPSET DATABASE format
'/data2/backup/dbf_%d_%t_%s.rman' TAG='DBF_CCEBI_20080620_1345'
```

Dosyaları sıkıştırma boyutu da oldukça muazzam; neredeyse %80'e varan bir alan kazanabiliyorsunuz. Sıkıştırılarak oluşturulan backupset'lerin normallerinden hiçbir farkı olmaz. Recover esnasında kullanılan komutlar ya da sorgulama aşamaları aynıdır. Tek farkı, işlemlerin daha yavaş yapılmasıdır. İster istemez sıkıştırılmış bir dosyanın içeriğinden veri çekmek daha fazla işlem gücü gerektirir.

#### 3.3. Alınan Yedeğe Arşiv Dosyalarını Eklemek

Daha önce arşiv dosyalarını yedeklemeyi görmüştük. Arşiv dosyalarını daha önce anlattığımız şekilde yedekleyebileceğiniz gibi, veritabanı yedeği alırken de yedekleyebilirsiniz. Böylece ayrı bir komut girmeden, tek adımda hem veritabanını yedekleyebilir, hem de arşiv dosyalarını bu yedeğe ekleyebilirsiniz. Bunun için aşağıdaki komutu kullanabilirsiniz;

```
RMAN> BACKUP DATABASE format '/data2/backup/dbf_%d_%t_%s.rman'
TAG='DBF_CCEBI_20080620_1345' PLUS ARCHIVELOG FORMAT
'/data2/backup/arc_%d_%t_%s.rman' TAG='ARC_CCEBI_20080620_1345';
```

#### 3.4. En Gelişmiş Yedek Alma Komutu

Şimdi daha kompleks bir işlem yapalım; yedek alalım, bu yedeğin içerisine arşiv dosyaları da eklensin ve bütün bu yedek sıkıştırılsın. **Aşağıdaki komut belki de en yararlı RMAN komutudur:**

```
RMAN> BACKUP as COMPRESSED BACKUPSET DATABASE FORMAT
'/data2/backup/dbf_%d_%t_%s.rman' TAG='DBF_CCEBI_20080620_1345' PLUS
ARCHIVELOG format '/data2/backup/arc_%d_%t_%s.rman'
TAG='ARC_CCEBI_20080620_1345';
```

Yaptığı işe göre oldukça basit duran bir komut. Gerçi bu komutu keşfetmem bir günümü almıştı :) Ancak RMAN ile çalışma yapacaksanız, Victorinox İsviçre Ordu çakınız işte burada!



### 3.5. Alınan Yedeklerin Hâlâ Kullanılabilir Olup Olmadığını Kontrol Etmek

Alınan yedeğe çok güvenmemek gerekiyor. Silinebilir, bozulabilir, erişilmez olabilir vs... Bu gibi durumları RMAN'in "**crosscheck backup**" komutuyla kontrol edebilirsiniz.

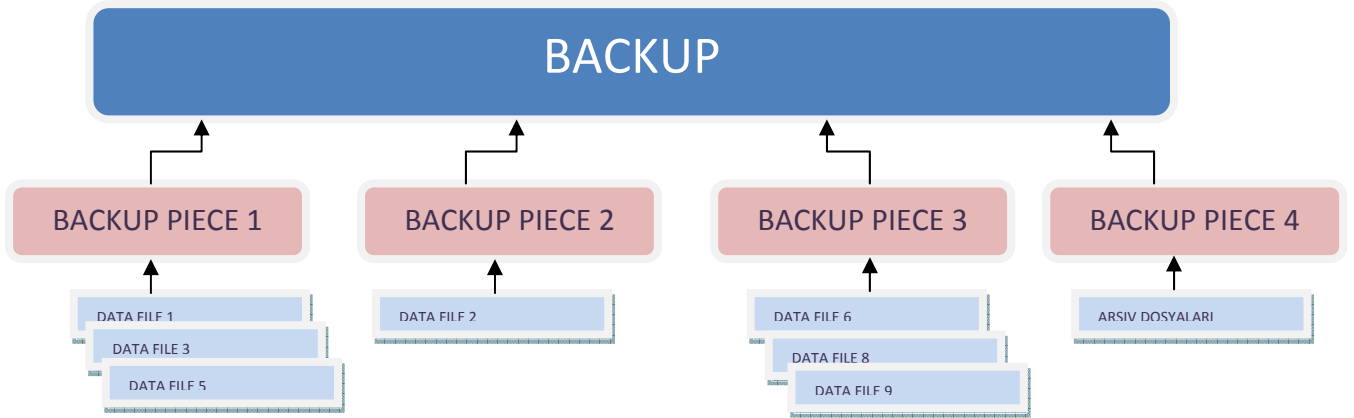
```
RMAN> crosscheck backup;
```

Bir yedek dosyasına artık erişilmezse; ilgili backup, expired duruma düşer ve kullanmayacağınız anlamına gelir.

### 3.6. Backup Piece Mantığı

Yedekleme işlemi başlattığınız zamanla, yedek işleminin tamamlandığı zaman arasında yeni arşivler oluşabilir. RMAN oldukça gelişmiş bir program ve bu duruma da bir çözüm üretmektedir. Sizin yedek almanız biterken, RMAN yeni arşiv dosyalarının oluşup oluşmadığına bakar ve eğer varsa, bunları yeni bir backup piece içinde aldığınız yedeğe ekler.

Backup piece'ler konusunda kafa karışıklığını önlemek için ne olduklarından bahsedelim. Bir backup, birden çok dosyadan oluşur ve bu dosyalara backup piece denmektedir. Bir backup piece içeriğinde, veritabanına ait dosyalar vardır. Bunlar datafile'lar ya da arşiv dosyaları olabilir.



RMAN'in bir başka güzelliği de, backup piece'leri kullanabilmenin esnekliğidir. Örneğin sadece belli başlı datafile'lara ihtiyacınız varsa, onu taşıyan backup piece'in var olması yeterlidir. Çok kötü bir senaryo gerçek olsa ve veritabanıyla birlikte RMAN yedeklerinizin bir kısmı da yok olsa, bir ihtimal bazı şeyleri bu sayede kurtarabilirsiniz.

### 3.7. Backup Piece'lerin Lokasyonunu Değiştirme

RMAN'in birçok şeyi oldukça basit gerçekleştirebildiğinden bahsetmiştik. Ancak her şeyi bu kadar basit gerçekleştirirken, esnekliğini de kaybeder. Örneğin oluşturduğunuz backup'ları farklı bir lokasyona taşırsanız, RMAN artık görmeyecektir. crosscheck backup komutunu çalıştırırsanız, dosyalar expired şeklinde gözükür.

Backup piece'leri farklı bir lokasyona taşırsanız; aşağıdaki adımları gerçekleştirerek, backup piece'leri RMAN'e tekrar tanıtabilirsiniz:

- Önce dosyaları kontrol ediyoruz:

```
RMAN> CROSSCHECK BACKUP ;
```

- Backup piece'ler bulunmadığı için expired konuma düşüyor. Artık bu dosyaları kullanmamız mümkün değil. TAG kullanarak ya da expired olma durumuna göre, bu yedekleri RMAN listesinden kaldırıyoruz:

```
RMAN> DELETE FORCE EXPIRED BACKUPSET OF DATABASE;
```

- Backup piece'lerin yeni lokasyonunu teker teker belirtiyoruz:

```
RMAN> CATALOG BACKUPPIECE '/data2/yeni_yedekler/BACKUPPIECE_ADI.RMAN'
```

Bu işlem sonrası list backup summary dersiniz backup piece'lerin RMAN backup'ları arasına eklendiğini göreceksiniz.

### 3.8. Konfigürasyon Ayalarını Değiştirme

RMAN'in birçok ayarını değiştirebilirsiniz. Değiştirilebilecek ayarları görebilmek için aşağıdaki komutu girebilirsiniz:

```
RMAN> SHOW ALL;
```

Karşınıza çıkacak sonuç, RMAN'in default gelen konfigürasyonunu göstermektedir. Her bir parameterenin RMAN'in çalışma yapısında etkisi bulunuyor. Ben en önemli bulduğum özellikleri aşağıda paylaşıyorum:

- İşlem yapılacak aygıtı disk olarak belirler:  
RMAN> CONFIGURE DEFAULT DEVICE TYPE TO DISK;
- Yedek alma optimizasyonunu açar:  
RMAN> CONFIGURE BACKUP OPTIMIZATION ON;
- Aynı anda RMAN için kaç process çalışacağını gösterir. Ne kadar yükseltirseniz, RMAN ile yedek işlemi o kadar çabuk biter ancak sistem o kadar çok yavaşlayacaktır. Aşağıdaki örnekte aynı anda paralel işlem sayısı 4 olarak belirlenmiştir. **(Bu özelliği kullanırken çok dikkatli olmak gerekiyor.)**

```
RMAN> CONFIGURE DEVICE TYPE DISK PARALLELISM 4;
```

Bu ayarlarda yaptığınız değişiklikleri daha sonra aşağıdaki gibi iptal edebilirsiniz:

```
RMAN> CONFIGURE DEFAULT DEVICE TYPE CLEAR;
```

```
RMAN> CONFIGURE DEVICE TYPE DISK PARALLELISM CLEAR;
```

```
RMAN> CONFIGURE BACKUP OPTIMIZATION CLEAR;
```

**CLEAR** özel bir anahtar kelimedir, '**default**' ayara dönmeye yarar. Hemen her ayarın ilk hâline dönmek için bunu kullanabilirsiniz.

## 4. YEDEK BİLGİLERİNİ GÖRME

RMAN ile alınan yedekler özel bir formatta saklanır. RMAN ile oluşturulan dosyaların ismine bakarak, hangi tarihe neye istinâden yedek alındığını bulmak biraz güç olabiliyor. Bunun için aşağıdaki komutları kullanmak uygun olacaktır.

- Alınan yedeklerin özet bilgisini görmek:  
RMAN> LIST BACKUP SUMMARY;
- Alınan yedeklerin tam olarak içeriğini görmek:  
RMAN> LIST BACKUP;
- Hangi dosyanın hangi yedek parçası (backup piece) içinde olduğunu görmek:  
RMAN> LIST BACKUP OF DATABASE;
- Bir önceki komutu belirli bir tarih aralığında girerek kullanmak:  
RMAN> LIST BACKUP OF DATABASE BETWEEN '01-ARP-2008' AND '28-JUN-2008'
- Alınan yedeklerin taşıdığı dosyaları görmek:  
RMAN> LIST BACKUP BY FILE;

Yukarıda verilen listeleme komutlarıyla sınırlı kalmanız gerekmez. Farklı kombinasyonlar deneyip, listeleme özelliklerinin ne kadar geniş olduğunu farrkedebilirsiniz.

## 5. YEDEKTEN DÖNME İŞLEMLERİ

Bu bölümde almış olduğumuz yedekten dönme işlemine bakacağız. Başlamadan önce bir uyarıda bulunmak gerekiyor: **Eğer hangi yedekten dönülmesi gerektiğini belirtmezseniz, son alınan yedek işleme konur.**

### 5.1. Yedek Dosyalarının Doğruluğunu Kontrol Etme

Aldığınız yedeklerin kontrolünü yapmak akıllıca bir yöntem olacaktır. Yedek aldığınızı düşünüp, aldığınız yedeklerin bozuk olduğu gerçeği ile karşılaşmak bir kriz anında büyük sıkıntılara sebep olacaktır. RMAN bu konuda oldukça güzel bir özellik sunuyor.

Yedekten dönmeden, sadece yedek dosyalarının yapısı incelenerek, yedek dönme işleminin başarılı olup olamayacağı RMAN tarafından bulunabilir. Bunun için aşağıdaki gibi yedeği **'validate'** etmek gerekir:

```
RMAN> RESTORE DATABASE VALIDATE ;
```

Bu komut sonucunda son alınan yedeğin çalışıp çalışmayacağı gösterilecektir. Eğer daha farklı bir backupset'i kontrol etmek isterseniz, aşağıdaki gibi etiket (TAG) parametresinden yararlanabilirsiniz:

```
RMAN> RESTORE DATABASE VALIDATE FROM TAG='CCEBI_YEDEK_2008_06_20' ;
```

### 5.2. Full Recover (Tam Yedekten Dönüş):

Database'in tamamen çökmesi ya da tamamen değişmesi durumunda full recover gerekecektir. Full recover için, **database'in mutlaka mount mode'da açılmış** olması gerekir. Veritabanını mount mode'da açıktan sonra yapılması gereken işlem oldukça basit;

```
RMAN> RESTORE DATABASE ;  
RMAN> RECOVER DATABASE ;
```

Yukarıdaki komutların çalıştırılması sonucunda, database kurulacaktır. Sadece iki satır komutla database'i recover edebilmek insanı yanılgıya düşürecek kadar kolay bir yöntem. Ancak dikkat edilmesi gereken bu komutları çalıştırdığınız takdirde, alınan son yedeğe göre recover edileceğidir. **FROM TAG** parametresini ekleyip, başka yedeklerin kullanılmasını sağlayabilirsiniz.

### 5.3. Datafile'ların ve Tablespace'lerin Recover Edilmesi

Her zaman full recover gerekmez. Bazı durumlarda, datafile ya da tablespace bozulur. Bu disk üzerindeki fiziksel bir hasardan gerçekleşebilir ya da dikkatsiz bir kullanıcısı datafile'a zarar verebilir. Bu gibi durumlarda database'i kapatmadan, ya da mount mode'a çekmeden ilgili tablespace veya datafile'ları recover edebilirsiniz. İşlemi gerçekleştirmek için dikkat edilmesi gereken en önemli konu, ilgili **datafile ya da tablespace'lerin offline mode'da** olmasıdır. Dilerseniz, database'i tamamen kapatıp, mount mode'a alıp, bu işlemleri yine gerçekleştirebilirsiniz. Ancak buna gerek yok. SYSTEM tablespace'i hariç, bütün tablespace'ler offline mode'a çekilerek işlem yapılabilir. Aşağıda datafile ve tablespace restore etmenin çeşitli yöntemlerini göreceksiniz:

- **Dosya lokasyonunu kullanarak datafile recover etmek:**  
RMAN> RESTORE DATAFILE '/u02/oradata/cuddle/users01.dbf';  
RMAN> RECOVER DATAFILE '/u02/oradata/cuddle/users01.dbf';

- **Dosya numarası (FILE#) kullanarak datafile recover etmek\* :**

```
RMAN> RESTORE DATAFILE 3,4,12;  
RMAN> RECOVER DATAFILE 3,4,12;
```

(\* Datafile FILE# değerini nasıl bulabileceğinizi, bölüm 2.3'te görebilirsiniz. )

- **Tablespace belirterek recover etmek:**

```
RMAN> RESTORE TABLESPACE USERS, DATA_TS;  
RMAN> RECOVER TABLESPACE USERS, DATA_TS;
```

Bu verilen örneklerin hepsi son backup'a göre recover edilir. Eğer daha farklı bir backup kullanmak istiyorsanız, **FROM TAG** ifadesini kullanabilirsiniz.

#### 5.4. Arşiv Dosyalarını Yedekten Dönme

Arşiv dosyalarını yedeklemek için muhtelif yöntemleri "[2.4. Arşiv Dosyalarını Yedeklemek](#)" bölümünde görmüştük. Benzer şekilde arşiv dosyalarının recover işlemi söz konusu ise birçok seçenek göz önüne getirilebilir.

- **SCN – System Change Number'a belirli aralıktaki arşiv dosyalarını dönme:**  
RMAN> RESTORE ARCHIVELOG SCN BETWEEN 26716797844 and 26716798848;
- **Yedeklenmiş bütün arşiv dosyalarını dönme:**  
RMAN> RESTORE ARCHIVELOG ALL;
- **30 günden yeni 7 günden eski arşiv dosyalarının yedeğini dönme:**  
RMAN> RESTORE ARCHIVELOG FROM TIME 'SYSDATE-30'  
UNTIL TIME 'SYSDATE-7';

Verilen örnekler en genel kullanılabilir yöntemleri gösteriyor. Daha başka çeşitlerde de (sequence numarası gibi...) yedekten dönme işlemlerini yapabilirsiniz.

#### 5.5. CONTROL FILE'i Yedekten Dönme

CONTROL FILE'in silineceği pek fazla durum olmayabilir; olmamalıdır da. Ancak böyle bir durum başınıza gelirse ve RMAN üzerinden control file'in yedeğini almışsanız aşağıdaki gibi control file'i restore edebilirsiniz. Tabii bunun için database'in nomount mode'da açılmış olması gerekir:

```
RMAN> STARTUP NOMOUNT;  
  
RMAN> RESTORE CONTROLFILE FROM '/data2/cf_1SF32433.rman';  
  
RMAN> ALTER DATABASE OPEN;
```

## 5.6. Until Time Şeklinde Yedekten Dönme

RMAN ile veritabanını istediğiniz bir zamana döndürebilirsiniz. Bunun için aşağıdaki gibi bir giriş yapmanız yeterlidir:

```
RMAN> RESTORE DATABASE UNTIL TIME
"TO_DATE('2008-06-25 16:45','YYYY-MM-DD HH24:MI')";

RMAN> RECOVER DATABASE UNTIL TIME
"TO_DATE('2008-06-25 16:45','YYYY-MM-DD HH24:MI')";
```

Until Time'i tablespace'lerde de kullanabilirsiniz. Ancak Until Time'i kullanırken aşağıdaki durumlara dikkat etmeniz gerekiyor:

1. Until Time ile veritabanını döndüreceğiniz zaman, yedek tarihinden sonra olmalı. Örneğin 16.30'da yedeği aldıysanız, 15.30'a dönemezsiniz. Mutlaka 16.30'dan sonraki bir zaman belirtilmeli.
2. Until Time ile dönüş yapabilmek için arşiv dosyalarının mevcut olması gerekmektedir. Until Time ile recover yaparsanız, önce arşiv dosyalarının olduğundan ve doğru konumda olduğundan emin olun.

## 5.7. REDO LOG'ların Olmamasının Sorun Yaratması Durumunda

RMAN ile until time şeklinde dönüş yaparken, bazı durumlarda sorunlarla karşılaştım. Özellikle redo log dosyalarının olmamasıyla alakalı sıkıntılar ortaya çıktı. Eğer buna benzer bir durumla karşılaşırsanız, aşağıdaki yöntemi izleyebilirsiniz:

- a. SQL ortamında, aşağıdaki ifadeyi çalıştırın ve control file oluşturma script'ini yaratın:

```
SQL> ALTER DATABASE CONTROLFILE TO TRACE AS '/tmp/create_cfile.sql';
```

- b. Oluşturulan control file yaratma script'ini düzenleyin.
- c. Database'i until time şeklinde tekrar recover etmeyi deneyin.

## 6. CATALOG MANTIĞI

Bu noktaya kadar bayağı bir şeyden bahsettik. Ancak kasıtlı catalog mantığını sona bıraktım. RMAN ile alınacak yedeklerin kaydını, bir veritabanında tutabilirsiniz. Hangi yedek ne zaman alınmış; hangi veritabanına aitmiş; hangi dosyalar, hangi backupset'ler içerisinde bulunuyor vs... Bunların hepsini CATALOG denilen özel bir veritabanı şemasında tutmanız mümkündür.

İnsanın aklına böyle bir yapının gereği var mı diye bir soru geliyor. Tek bir veritabanıyla uğraşmanız durumunda gerek görülmeyebilir. Hatta ekstra bir külfet olarak değerlendirmeniz bile mümkündür. Fakat veritabanı sayısı arttıkça merkezi bir noktadan hepsini kontrol edebilmek için CATALOG yapısı gerekmektedir. CATALOG kullanmadığınız takdirde alınan yedeklere dair bilgiler ilgili veritabanının **CONTROL FILE** dosyasında binary olarak tutulur.

Şimdi bir catalog oluşturup, bir veritabanını bu catalog'a dahil edelim ve sonrasında veritabanının yedeğini alalım.

- a. Önce catalog şemamızı tutacak merkezi bir veritabanı belirliyoruz ve aşağıdaki ifadeleri SQL üzerinden çalıştırıyoruz:

```
SQL> CREATE USER rman IDENTIFIED BY cat
      TEMPORARY TABLESPACE temp
      DEFAULT TABLESPACE tools
      QUOTA UNLIMITED ON tools;
```

- b. Şimdi rman isimli kullanıcıya hak catalog için hak veriyoruz:

```
SQL> GRANT RECOVERY_CATALOG_OWNER TO rman;
```

- c. SQL tarafında yapmamız gerekenleri tamamladık; şimdi RMAN programını açıyor ve aşağıdaki komutları girerek CATALOG'u yaratıyoruz. Burada dikkat etmeniz gereken, CATALOGDB diye geçen ifadenin TNSNAMES.ORA'daki bir tanım olduğudur. Siz hangi database'i catalog için kullanacaksınız, onun TNSNAMES.ORA'daki tanımını kullanabilirsiniz:

```
$ rman
RMAN> CONNECT CATALOG rman/cat@CATALOGDB;
RMAN> CREATE CATALOG TABLESPACE cat_ts;
```

İşlem sonucunda "**recovery catalog created**" şeklinde bir ifade gelecektir. Artık RMAN'i catalog ile kullanabilirsiniz. CATALOG'a bağlanmak için aşağıdaki komutu kullanmanız gerekmektedir:

```
RMAN> CONNECT CATALOG RMAN/CAT@CATALOGDB;
```

( **NOT:** rman'in kullanacağı tablespace'i farklı bir şey de yapabilirsiniz. Ancak tablespace adını catalog olarak vermeye kalkarsanız, tek tırnak içinde 'catalog' şeklinde belirtmeniz gerekmektedir. )

- d. CATALOG olarak kullanacağımız şemayı yarattık. Şimdi sıra bir veritabanına bağlanıp, bunu RMAN CATALOG'a dâhil etmektir. Bunun için veritabanına ve catalog'a bağlanmamız gerekir. Aşağıdaki işlemi önce Unix komut satırından yapıyoruz:

```
$ rman target sys/password@yedeklenecek_db  
catalog rman/cat@CATALOGDB;
```

Bu komutla hem yedeklenecek veritabanına bağlandık; hem de yedekleme bilgilerinin tutulacağı catalog şemasına... Bundan sonra veritabanını catalog'a aşağıdaki gibi kaydediriz:

```
RMAN> REGISTER DATABASE;
```

Artık daha önce gördüğünüz komutların hepsini kullanabilir ve yedek alma işlemini sorunsuzca gerçekleştirebilirsiniz.

Catalog yapısı işleri biraz daha karmaşıklaştırmış gibi gözükebilir. Fakat çoklu veritabanı söz konusu ise tercihinizi mutlaka catalog'tan yana kullanın. En basit avantajı, yedek bilgilerinin controlfile'da tutulmamasıdır. Yarın öbür gün veritabanının bozulmasıyla birlikte controlfile da silinirse, işleriniz çok daha karmaşıklaşacaktır.



## 7. DEĞERLENDİRME

RMAN birçok işi otomatize ediyor ve DBA üzerindeki yedekleme külfetini büyük ölçüde hafifletiyor. Ancak işleri bu kadar basit yapabilmek, esnekliği bir ölçüde azaltıyor ve bu yüzden mucizeler beklenmemesi gerekiyor.

Mevcut bir yedek alma düzenine RMAN'ı dahil etmek yerine sıfır kurulacak ortamlar için RMAN'ı değerlendirmek gerekir. Yedeklerin daha derli toplu alınması, daha kontrolü olması ve daha az yer kaplaması nedeniyle birçok ortam için oldukça yararlı olacaktır. Fakat varolan yapıyı bozmaktansa, sıfırdan devreye girecek noktalarda RMAN'ı kullanmamız gerektiğine inanıyorum.