

---

# AULA 10

---

**MÓDULO DBA PLENO**

DISASTER RECOVERY



# Na aula anterior...

---

Aprendemos mais sobre backup e algumas formas de automatizá-lo.

Aprendemos dicas importantes e valiosas sobre RMAN, CRONTAB e agendamentos.

# Na aula de hoje

---

Vamos aprender que poderemos fazer além do backup que aprendemos, o backupset, o backup copy e também aprenderemos algumas técnicas de se fazer backup lógico.

Aprender a diferença básica entre eles e aprender onde poderemos utilizar um ou outro.

# Tipos de BACKUP

## Lógico e físico

---

Nas aulas anteriores conversamos a respeito do backup feito com o RMAN. Este tipo de backup é também conhecido como backup físico, pois ele conterà cópia dos datafiles em questão, ou seja, ele faz cópia do datafile físico.

Esta cópia pode ser feita basicamente de duas formas: **Backupset** ou **Backup copy**.

Ambas são eficientes, porém tem uma diferença básica e que algumas vezes é cobrada em provas de certificação e que alguns acabam tendo dúvidas.



# Backup Físico

## Backupset

---

Quando eu faço um backup do tipo **backupset** os dados do meu backup serão armazenados em peças de backup. Dentro de uma peça de backup por exemplo poderá conter mais de um datafile.

Backupset 01
Datafile USERS
Datafile SYSTEM
Datafile SYSAUX
...

Backupset 02
Datafile EXAMPLE
Datafile UNDO
Datafile APP_DATA
...

Assim teremos diversos backupsets e dentro deles os dados dos datafiles. É a forma de backup mais utilizada, e segura, levando em consideração que ela cria as peças de backup e estes são binários lidos apenas pelo Oracle.

# Backup Físico

## Backup Copy

---

Quando eu faço um backup do tipo **Backup Copy**, com o comando *backup as copy* os dados serão copiados de forma muito parecida com o copiar colar.

Criando cópia física dos mesmo.

Backup copy é muito útil em diversas situações, mesmo não sendo o mais usado.

Vamos ver agora um exemplo de backup copy.



# Backup Copy

## Utilização prática

---

Poderemos utilizar um backup copy para trazer de forma bem simples um datafile de filesystem para ASM. Veja abaixo a sequência de comandos que fizemos:

**1- Criamos um backup cópia já no formato do ASM:**

```
RMAN> BACKUP AS COPY FORMAT '+DATA' datafile 5;
```

**2- Conferimos o resultado:**

```
RMAN> list copy of datafile 5;
```

**3- Colocamos o datafile (apenas o datafile) offline:**

```
RMAN> sql 'alter database datafile 5 offline';
```

**4- Indicamos ao banco a troca de datafile (localização e arquivo físico) escreve o controlfile:**

```
RMAN> switch datafile 5 to copy;
```

**5- Procedemos o recovery para caso necessitasse:**

```
RMAN> recover datafile 5;
```

**6- Recolocamos o datafile online e pronto para utilização**

```
RMAN> sql 'alter database datafile 5 online';
```

# Backup Lógico

## Datapump

---

Em diversas situações é mais interessante a utilização de um backup lógico. Isso, porque ele é mais rápido de fazer, mais simples de retornar e ocupa menos espaço que um backup completo do banco de dados.

Porém, nem tudo são flores, não são todas as situações onde ele será a melhor saída.

Estou falando do DATAPUMP. Ferramenta muito interessante da Oracle e que sem dúvida é muito útil.

Sua utilização normalmente é para se fazer cópia de um schema(owner) específico ou grupo deles.



# Datapump

## Algumas utilizações

---

O datapump é uma ferramenta que me permite fazer um export/import de determinado owner no banco de dados.

É comum criarmos um script que pode ser adicionado na crontab ou executado via JOB do Oracle para fazer cópia lógica de determinados owners.

Porém, sua limitação está no fato de que ele quando precisar ser restaurado, não existirá aplicação de archive sobre o arquivo retornado, ou seja, voltando um DUMP (como chamamos os arquivos criados pelo datapump, os dados estarão como no ato da criação do export.

# Datapump

## Algumas utilizações

---

Antes de se proceder o export com o datapump através do comando expdp necessita antes de algumas verificações/configurações:

1- `CREATE DIRECTORY BKPS AS '/u01/app/oracle/bkp/';`

2- Após criar o Diretório precisamos conceder os privilégios de leitura e gravação. Em nosso exemplo iremos disponibilizar este diretório para todos os usuários do Banco de Dados concedendo o privilégio para o schema Public;

`GRANT read, write ON DIRECTORY BKPS TO PUBLIC;`

3- Iremos conceder o Privilégio `IMP_FULL_DATABASE` para o usuário teste que irá Importar:-  
`GRANT IMP_FULL_DATABASE to teste;`

# EXPDP/IMPDP Comandos

```
SET PAGESIZE 100
SET LINESIZE 350
SELECT 'expdp system/<senha> dumpfile=expdp_'
  ||username
  ||'_ ' || to_char(SYSDATE,'DD') || to_char(SYSDATE,'MM') || to_char(SYSDATE,'YYYY') || '.dmp schemas='
  ||username
  ||' directory=BKPS logfile=expdp_'
  ||username
  ||'_ ' || to_char(SYSDATE,'DD') || to_char(SYSDATE,'MM') || to_char(SYSDATE,'YYYY') || '.log '
  || 'FLASHBACK_TIME=\''TO_TIMESTAMP\(' ' || to_char(SYSDATE,'DD/MM/YYYY HH24:MI:SS') || '\',\''DD/MM/YYYY HH24:MI:SS\''\)'
FROM dba_users
WHERE USERNAME IN ('RAUL')
UNION
SELECT 'impdp system/<senha> dumpfile=expdp_'
  ||username
  ||'_ ' || to_char(SYSDATE,'DD') || to_char(SYSDATE,'MM') || to_char(SYSDATE,'YYYY') || '.dmp schemas='
  ||username
  ||' directory=BKPS logfile=expdp_'
  ||username
  ||'_ ' || to_char(SYSDATE,'DD') || to_char(SYSDATE,'MM') || to_char(SYSDATE,'YYYY') || '.log'
FROM dba_users
WHERE USERNAME IN ('RAUL');
```

# EXPDP/IMPDP

## Comandos

---

O Script facilita a criação automática dos comandos de export e import posteriormente.

O expd e impdp veio substituir os antigos exp e imp trazendo diversas vantagens de performance e de utilização através de diversos parâmetros.

Vamos ver na prática esta execução.

# Na Próxima aula

---

Vamos iniciar a correção de problemas causados por falhas humanas, de sistema ou de hardware.

Iniciaremos processos de recovery simulando falhas diversas no ambiente

---

# AULA 10

---

**fim**

