

Export e Import Oracle

Observação: Nesse exemplo irei usar orcl11 como o nome da instancia do banco para facilitar o entendimento

Iniciar a sessão no putty da máquina de origem.

Digite o comando “ps -ef | grep pmon” para ver as instancias que estão em execução

Após ver as instâncias, basta setar a variáveis de ambiente para o banco que deseja fazer o export. Digite “. oraenv orcl11”

Digite o comando do export sem está no root para evitar erro de privilégio

```
expdp \ "SYS@ORCL11 AS sysdba\ " directory=dumpdir dumpfile=exp_20022020_3.dmp  
schemas=ctis logfile=exp_20022020_3.log exclude=statistics content=metadata_only  
version=11.2
```

Após digitar o comando de export você precisa informar a senha do usuário que está tentando executá-lo. Neste caso é o SYS. Marquei em vermelho na instrução acima

Atenção no parametron version do export. É necessário verificar com a versão do banco de destino. Caso a versão seja inferior a do banco de origem, você precisa informar qual versão do BD que está realizando o export. Caso contrário, basta remover esse parametro.

Se precisar ver a versão do BD basta fazer o seguinte: select * from v\$version

Caso o export tenha sido realizado com sucesso e você precise importá-lo em outra máquina, faça a transferência através do comando scp.

```
scp EXP_20022020_3.dmp oracle@host:/u01/oradata/exports (substituir pelo endereço de  
origem e destino respectivamente)
```

O S.O vai pedir a senha da máquina de destino.

É importante ver se tem espaço para realizar o import antes de iniciar a instrução de import.

- Criar o usuário
- Criar tablespace
- Dar grant ao usuário nas tablespace

Feito isso você precisa conectar na máquina que será feito o import.

No nosso exemplo o diretório para onde enviamos o arquivo .dmp do export é:
/u01/oradata/exports

Você pode digitar o comando ls -lrt para verificar se o arquivo de fato está lá.

Caso você queira fazer o import a partir de um arquivo parfile, primeiramente você precisa criá-lo. Digite o comando **vi nome_arquivo.par**. Copie e coloque as instruções do import dentro desse arquivo

```
DIRECTORY=dumpdir
SCHEMAS=ctis REMAP_SCHEMA=ctis:CTIS_REMAP_TBS
REMAP_TABLESPACE=TOOL_DATA:CTIS_DATA,USERS:CTIS_DATA,DATA:CTIS_DATA,MVLOG_DATA:CTIS_DATA,
SYSTEM:CTIS_DATA,CTIS_TAB_MED:CTIS_DATA,INDX:CTIS_DATA
~
~
~
~
~
~
~
~
~
~
```

Feito isso salve o arquivo usando o comando ":wq + enter"

Feito isso digite a instrução para iniciar o import

```
$ impdp DUMPFILE=EXP_20022020_3.DMP logfile=IMP_20022020_RMP.log
parfile=nome_arquivo.par
```

Após finalizar o import verifique a quantidade de erros e olhe o log gerado.

A query abaixo analisa a quantidade de objetos criado na origem e destino para verificar se todos os objetos foram criados.

```
SELECT o2.*,
       (SELECT COUNT (1)
        FROM dba_objects o1
        WHERE o1.owner = 'BD1'
              AND o1.object_type = o2.object_type)
       qtd_destino
FROM ( SELECT object_type, COUNT (1) qtd_origem
      FROM dba_objects@SSZD_SBI
      WHERE owner = 'BD2'
      GROUP BY object_type) o2
ORDER BY QTD_ORIGEM - QTD_DESTINO DESC
```

Quantidade que tem do tipo de objeto em cada banco por owner

```
SELECT 'bd1', OBJECT_TYPE, COUNT(1) FROM DBA_OBJECTS WHERE OWNER = 'bd1' GROUP
BY OBJECT_TYPE
UNION
SELECT 'bd2', OBJECT_TYPE, COUNT(1) FROM DBA_OBJECTS@SSZD_SBI WHERE OWNER =
'bd2' GROUP BY OBJECT_TYPE ORDER BY 2
```

Trazer tudo que tem no primeiro que não tem no segundo

```
SELECT OBJECT_NAME FROM DBA_OBJECTS@SSZD_SBI WHERE OWNER = 'bd1' AND
OBJECT_TYPE = 'TABLE'
MINUS
SELECT OBJECT_NAME FROM DBA_OBJECTS WHERE OWNER = 'bd2' AND OBJECT_TYPE =
'TABLE'
```