

오라클 백업 및 복구(Export, Import)

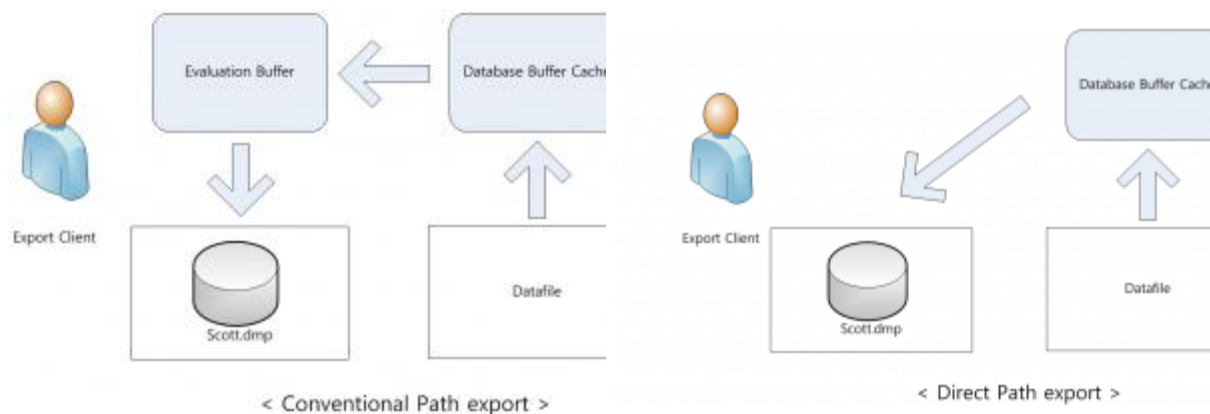
1. export / import

- 가. 여러개의 테이블 중에서 특정 table 만 백업/복구 하고자 할 때
- 나. 오라클의 버전, 플랫폼이 서로 다른 상황에서의 서버간 데이터 이동 시(migration)

2. export 방식

가. **Conventional Path export** : Evaluation Buffer 를 사용하는 방식, DB Buffer cache 에서 필요데이터를 Evaluation Buffer 로 복사 후 데이터를 가공(text -> binary)하여 디스크에 파일로 저장함. export 작업 중에 발생하는 DDL, DML 등의 명령들은 백업파일에 반영되지 않는다.(백업 파일은 Evaluation Buffer 을 이용하여 작업하기 때문)

나. **Direct Path export** : DB Buffer Cache 에서 데이터를 가공(text -> Binary)하여 디스크에 파일로 저장함, export 명령 이후에 백업대상이 되는 테이블스페이스나 테이블에 Lock 이 발생하기 때문에 DDL, DML 작업은 실패 또는 보류 된다.



3. export 옵션 및 사용예제

가. 옵션

- userid/passwd : export 를 수행하는 계정/패스워드
- buffer : Evaluation Buffer 크기 지정(용량이 클 수록 export 작업이 빨라진다)
- file : export 결과를 저장할 파일명
- full : 전체 DB 를 export 할 것인가 지정
- owner : export 받을 사용자 이름지정
- tables : export 받을 테이블 이름 지정
- tablespaces : export 받을 테이블스페이스 이름지정
- parfile : export 옵션을 미리 지정한 파라미터 파일지정

나. 사용예제

```
exp system/oracle full=y file=/backup/export/test01.dmp
```

```
exp system/oracle full=y file=/backup/export/test02.dmp direct=y
```

```
exp system/oracle tables=emp W
```

```
file=('/backup/export/test03_1.dmp', '/backup/export/test03_2.dmp') filesize=10M
```

```
exp system/oracle tablespaces=(example, undotbs1) file=/backup/export/test04.dmp

exp system/oracle file=/backup/export/test05.dmp owner=(scott, hr)

exp system/oracle file=/backup/export/test06.dmp full=y buffer=1024000

vi par_full.dat
file=/backup/export/test07.dmp
full=y
direct=y

exp system/oracle parfile=par_full.dat

exp scott/tiger query=W"where ename like W'F%W'W" tables=emp W
file=/backup/export/test07.dmp
```

4. import 옵션 및 사용예제

가. 옵션(export의 옵션과 유사하다)

- userid/passwd : import 를 수행하는 계정/패스워드
- buffer : Evaluation Buffer 크기 지정(용량이 클 수록 import 작업이 빨라진다)
- full : export 파일의 모든 데이터를 import 한다.
- file : import 할 export 파일명 지정
- show : 데이터를 import 하지 않고 내용만 확인함
- ignore : import 작업 중 발생할 수 있는 에러를 무시하고 다음단계의 작업을 진행함
- fromuser : export 할 당시의 object의 소유자 지정
- touser : import 할 object의 새 소유자 지정
- tables : import 할 테이블 이름 지정
- parfile : import 옵션을 미리 지정한 파라미터 파일지정

나. 사용예제

```
imp system/oracle file=/backup/export/test01.dmp ignore=y full=y

imp system/oracle file=/backup/export/test02.dmp W
fromuser=scott touser=hr ignore=y

imp system/oracle file=/backup/export/test03.dmp full=y show=y log=test03.log
```

참고 : export/import 계정

import 할 때 사용하는 계정은 export 할 때 사용한 계정이어야 한다. 이 계정이 같지 않으면 import 수행 시 오류가 발생한다. 만일 export 계정을 잊었다면 덤프파일을 vi 편집기로 열어 확인할 수 있다.(2 번째 줄)

참고 : import 작업 중 에러발생 시

import 작업을 진행하던 도중 에러가 발생해 같은 작업을 반복하게 되면, import 대상이 되는 테이블(제약조건이 없는)에 데이터가 중복 저장될 수 있다. 그러므로 같은 작업을 반복시에는 import 대상이 되는 테이블의 내용을 지우고(drop 또는 truncate) 진행해야 한다.

참고 : SYS 계정으로 생성된 Object export

일반적으로 SYS 계정에서 생성된 객체는 export 명령어로 백업할 수 없으므로 주의해야 한다.
(단, 경우에 따라서 system 계정으로 백업이 가능하기도 하다)

4. Import 대상 서버에서 필요한 사전 작업

- 가. Export 한 서버와 동일한 Tablespace 생성
- 나. 충분한 크기의 Temporary Tablespace 확보
- 다. Export 한 서버와 동일한 사용자 생성

```
exp userid=system/1234@xe owner=clamp file=c:\W140820.dmp
```

```
imp system/1234@xe fromuser=clamp touser=clamp  
file=c:\W140820.dmp
```

–백업(Export) :데이타와 구조를 바이너리 파일로 저장

```
dos>exp userid=아이디/비밀번호@전역데이타베이스명(xe)  
file=저장경로
```

예]

1. system 계정으로 전체 백업

```
dos>exp userid=system/비밀번호@전역데이타베이스명 full=y  
file=c:\Wdump.dmp
```

2. system 계정으로 scott 계정에 있는 DB 백업

```
dos>exp userid=system/비밀번호@전역데이타베이스명  
owner=scott file=c:\Wdump.dmp
```

3. scott 계정으로 자신의 모든 데이타 백업

```
dos>exp userid=scott/비밀번호@전역데이타베이스명  
file=c:\Wdump.dmp
```

4. scott 계정으로 emp 테이블만 백업

```
dos>exp userid=scott/비밀번호@전역데이타베이스명  
file=c:\Wdump.dmp tables=emp
```

*여러개 테이블을 동시에 받으려면 tables=(테이블 1,테이블 2,...)

*백업 파일의 확장자는 보통 .dmp 혹은 .dat .bak 으로 한다 한다.

-복원(Import)

imp 아이디/비밀번호@전역데이터베이스명 file=백업경로

예]

1. system 계정으로 전체 복원

```
dos>imp system/비밀번호@전역데이터베이스명  
file=c:\Wdump.dmp
```

2. system 계정으로 scott 계정에 있는 DB 복원

```
dos>imp system/비밀번호@전역데이터베이스명 fromuser=scott  
touser=scott file=c:\Wdump.dmp
```

3. :scott 계정으로 자신의 모든 데이터 백업

```
dos>imp scott/비밀번호@전역데이터베이스명  
file=c:\Wdump.dmp
```

4. :복원하고자하는 DB 에 같은 이름의 Object 가 있을때,오류를 무시하고 건너 띄고 싶을때 ignore 옵션사용

```
dos>imp 아이디/비밀번호@전역데이터베이스명  
file=c:\Wdump.dmp ignore=y
```

5. system 계정으로 들어가 scott 에서 Export 한 데이터를 scott2 에게 Import

```
dos>imp system/비밀번호@전역데이터베이스명 fromuser=scott  
touser=scott2 file=c:\Wdump.dmp
```

오라클백업

knowhow/Oracle 2008/08/22 11:06

오라클백업에는크게피지컬백업(Physical Backup) 방식과로지컬백업(Logical Backup) 방식이있다.
피지컬백업방식이란데이터파일자체를가피하는방식을말하는데,
이것은다시데이터베이스를셋다운하고백업하는방식과오픈된상태에서백업하는방식으로나뉜다. 반면,
로지컬백업방식이란오라클에서제공하는유틸리티중의하나인 **Export/Import** 를이용하여백업하는형식으로,
이백업파일을이용하여리커버리를실행하면 **Create Object** 문장을이용해테이블스페이스와유저,
테이블을생성하고, **Insert Statement** 를이용해특정테이블에 **Insert**
작업이이루어지기때문에이처럼일컫는것이다.
먼저이들에대해각각알아보고, 데이터복구(**Recovery**)를위한기본개념들을다루고자한다.
그리고리커버리에대한구체적인설명은다음호에서다룰예정이다.

Cold Backup 받는방법

콜드백업(**Cold Backup**)은데이터베이스가정상적으로셋다운된상태에서데이터파일, 로그파일,
컨트롤파일을모두백업받는것을말한다.
셋다운하지않고오픈된상태에서백업을받으면백업받은내용을나중에사용할수가없으므로유의해야한다.
콜드백업을위해서데이터베이스를셋다운할때에는 **Normal** 또는 **Immediate** 옵션을사용해야하며,
Abort 를사용해서는안된다. 상황이여의치않아 **Abort** 를사용한경우에는셋다운후에다시스타트업하고
Normal 로셋다운한후백업을받도록한다.
우선이들파일을백업하기위해컨트롤파일과데이터파일및로그파일의위치를확인하여이들을 **tar, cpio, dd**
등의명령을이용하여백업받도록한다. NT에서는 **Copy** 명령이나탐색기를이용해서백업을받으면된다.

■컨트롤파일확인방법

sqlplus 에 **system/manager** 로접속후

```
SQL>select value from v$controlfile;
```

■데이터파일확인방법

```
SQL>select name from v$datafile;
```

■로그파일확인방법

```
SQL>select member from v$logfile;
```

이처럼각각의파일위치를확인하여위에서확인된 **3** 종류의파일들을백업받으면된다.
이때오라클은반드시셋다운된상태이어야한다. 항상은아니지만필요에따라서는패러미터파일까지받아두는것이좋
다. 패러미터파일은 **\$ORACLE_HOME/dbs/init.ora** 에있는데,
이초기화패러미터파일에기술되어진패러미터는시스템의성능에주요한역할을하므로이파일이없어진경우에예전의
성능을유지하려면이파일을필수적으로백업해야한다.

NT에서는 **C:\ORANT\DATABASE\init.ora** 파일이다.

일반적으로 NT의 **SID** 가처음만들어질때 **ORCL** 이므로이초기패러미터파일역시 **initORCL.ora** 라고되어있다.
여기서는오라클인스턴스이름으로환경변수 **ORACLE_SID** 로지정되어있다. 이 **ORACLE_SID** 는 Unix의경우
\$env, NT의경우는 **Registry** 에지정되어있으므로직접확인해볼수있다.

Hot Backup 받는방법

핫백업(**Hot Backup**)이란데이터베이스를셋다운하지않고백업하는방식으로,
일반적인경우는적용되지않고자신의데이터베이스를 **Archive**
Mode 로운영하는경우에만가능하다. 콜드백업을이용하여데이터를복구하는경우에는콜드백업을받은시점까지만
복구가가능한데반해, 이 **Archive**
방식으로데이터베이스를운영하는경우에는사용자가원하는시점까지데이터를복구할수있다. 즉,
원하는시간을주어서, **SCN** 번호를주어서, 혹은어느데이터파일한개만의복구도가능하다.

데이터 복구만 생각한다면 모든 데이터베이스를 Archive 방식으로 운영하는 것이 좋겠지만, 이 방식을 사용하는 경우 데이터베이스 관리자가 백업/리커버리에 관하여 많은 지식과 경험을 갖춰야 하며, 또 Archive Log 를 계속 유지 관리하는 데는 스페이스와 성능의 부하가 따르기 때문에 사용자의 적절한 판단에 의하여 이용하도록 한다. 그러면 Archive Log Mode 의 운영 방법에 대해 알아보자.

Archive Log Mode 운영방법

오라클에서 데이터베이스가 켜진 상태에서 백업을 받거나 문제가 생긴 시점까지의 완벽한 리커버리 작업을 수행하기 위해서는 데이터베이스를 아카이브 로그 모드로 운영해야 한다.

아카이브 로그 모드로 운영하기 위해서는 다음의 절차에 따라 변경해야 한다.

1. initSID.ora 파일과 configSID.ora 에 다음의 패러미터가 이미 설정되어 있는지 확인한 후에 없을 경우 initSID.ora 에 설정한다.

(1) LOG_ARCHIVE_START = TRUE

이 패러미터에 의해 데이터베이스가 처음 구동시 ARCH 프로세스가 기동.

Log Switch 발생시 Automatic Archive 를 수행한다.

만약 이 패러미터가 False 이면 Manual Archive 를 수행한다.

(2) LOG_ARCHIVE_DEST = /home/oracle8/dbs/archive_file/arc

아카이브 파일이 생성되는 위치인 디렉토리 및 확장자를 포함하지 않는 파일명을 지정.

여기에서 archive_file 까지는 디렉토리이며 마지막에 있는 arc 는 아카이브 로그 파일 이름의 첫 부분이다.

이 아카이브 로그 모드를 설정하면 로그 파일의 스위치가 일어날 때마다 지속적으로

Log_Archive_Dest 에 지정된 위치로 로그 파일이 쏘아게 되는데,

만일 부주의에 의해 아카이브 로그 파일이 깨지거나 분실되는 경우는 그 이후의 데이터 복구가 불가능하므로 주의하여야 한다.

이러한 문제를 줄이기 위해 Oracle8 부터는 아카이브 로그 파일을

Mirroring 하기 위하여 여러 군데에 보관할 수 있는 방안도 도입되었다. 이는

log_archive_duplex_dest 를 설정함으로써 가능하다.

또한 V8i 부터는 최대 5 군데에 이 로그 파일을 저장할 수 있도록 하여

LOG_ARCHIVE_DEST_1="LOCATION=/oracle8/arch_1 MANDATORY"

LOG_ARCHIVE_DEST_2="LOCATION=/oracle8/arch_2 OPTIONAL?"

처럼 지정이 가능하다

(3) LOG_ARCHIVE_FORMAT = %s.log

아카이브 파일의 확장자와 로그 시퀀스 번호의 형식을 지정.

이는 (2)에서 정의된 아카이브 로그 파일의 첫 부분 이름과 함께 나타난다.

arc123.log, arc124.log (123 과 124 는 로그 시퀀스 번호) 와 같은 형태의 파일이 지정된 위치에 생성된다.

2. 다음과 같이 작업하여 아카이브 로그 모드로 변환한다.

```
$ svrmgrl
```

```
SVRMGR> connect internal
```

```
SVRMGR> startup mount - ■
```

```
SVRMGR> alter database archivelog; - □
```

```
SVRMGR> archive log list - ●
```

```
Database log mode ARCHIVELOG - ◎
```

```
Automatic archival ENABLED - ◇
```

```
Archive destination ?/dbs_ar/offline_log/offline - ◆
```

```
Oldest online log sequence 123 - △
```

```
Next log sequence to archive 125 - ▲
```

```
Current log sequence 125 - ▷
```

```
SVRMGR> alter database open; - ►
```

■ DB 를 Startup Mount 까지만 한다.

□ 이 커맨드를 이용하여 아카이브 모드로 데이터베이스를 변경한다.

- 아카이브모드로변경되었는지를확인한다.

©데이터베이스가아카이브모드임을나타낸다. 만약

NOARCHIVELOG 로되어있으면변경되지않은것을의미한다.

◇ initSID.ora 파일에서 LOG_ARCHIVE_START 패러미터를 TRUE 로정의하였음을나타내며 False 인경우에는 DISABLED 로나타난다.

◆ initSID.ora 파일의 LOG_ARCHIVE_DEST 패러미터에서정의한아카이브할장소이다.

△ 3 개의 Redo Log 증가장오래된리두로그의시퀀스가 123 임을의미한다.

▲ 다음에아카이브받을파일의로그시퀀스번호를나타낸다.

▷ 현재사용중인리두로그의시퀀스가 125 임을의미한다.

만약이전부터아카이브로그모드로운영중이었다면여기에서아카이브로그파일은로그시퀀스 124 까지아카이브되어있다는것을의미한다.

▶ 아카이브모드로변경후 DB 를오픈한다.

No Archive Log Mode 로전환하는방법

반대로, Archivelog Mode 에서 No Archivelog Mode 로전환하는방법은다음과같다.

먼저, 위에서세팅했던 initSID.ora 파일과 configSID.ora 에있는다음패러미터앞에 #을넣고저장한다.

```
#LOG_ARCHIVE_START = TRUE
```

```
#LOG_ARCHIVE_DEST = /home/oracle8/dbs/archive_file/arc
```

```
#LOG_ARCHIVE_FORMAT = %s.log
```

```
$ svrmgrl
```

```
SVRMGR> connect internal;
```

```
SVRMGR> shutdown immediate
```

```
SVRMGR> startup mount
```

```
ORACLE instance started.
```

```
Database mounted.
```

```
SVRMGR> alter database noarchivelog;
```

```
Statement processed.
```

```
SVRMGR> alter database open;
```

```
Statement processed.
```

Hot Backup 하는방법

데이터베이스를셴다운하지않고데이터파일을백업하는방법이다.

이방법은콜드백업보다더복잡하지만데이터베이스가오픈되어있는도중에할수있고또한테이블스페이스별로백업할 수있다는장점이었다.

핫백업은항상데이터베이스를아카이브로그모드상태로두고실시한다.

| 방법 | 테이블스페이스단위로백업을실시한다.

```
svrmgrl> CONNECT INTERNAL
```

```
svrmgrl> ALTER TABLESPACE SYSTEM BEGIN BACKUP;
```

시스템테이블스페이스의모든데이터파일에대해서 OS Backup 한다.

```
$ tar cvf /dev/rmt0 /usr/oracle8/dbs/syst1ORA8.dbf
```

```
$ tar cvf /dev/rmt0 /usr/oracle8/dbs/syst2ORA8.dbf
```

```
svrmgrl> ALTER TABLESPACE SYSTEM END BACKUP;
```

```
svrmgrl> ALTER TABLESPACE USERS BEGIN BACKUP;
```

```
$ tar cvf /dev/rmt0 /usr/oracle8/dbs/user1ORA8.dbf
```

```
svrmgrl> ALTER TABLESPACE USERS END BACKUP;
```

다른테이블스페이스에대해서도같은방법으로한다.

주의할점은 BEGIN 과 END 사이에는해당테이블스페이스의데이터파일헤더정보가변경되지않는다는것이다.

따라서 OS 백업이종료됨과동시에 'ALTER TABLESPACE ... END BAKCUP'

커맨드를 실행하여 데이터 파일의 헤더가 변경되도록 한다.
이 아카이브 백업 파일을 이용한 복구 방법은 다음 기회에 다루도록 한다.

Export Backup 하는 방법

오라클에서 제공되는 **Export** 유틸리티는 데이터베이스에 저장된 데이터를 바이너리 형태의 **OS** 파일로 만들고,
필요시 **Import** 유틸리티를 이용하여 데이터베이스로 다시 올리는 방식이다 .

이 유틸리티는 각 오브젝트 단위로 처리가 가능하기 때문에 테이블 몇 개만을 복구한다든가 특정 사용자의 테이블들을 다른
테이블 스페이스로 옮긴다든가 또는 전체 데이터베이스의 자료를 서로 다른 **OS** 로 옮기는 경우 등에 특히 유리하다.

Export 의 종류

이 **Export** 방안은 시스템과 서로 메시지를 주고받으면서 백업을 할 수 있는 **Interactive Mode** 가 있고,
한 라인의 명령어로 **Export** 시 필요한 옵션을 길게 열거해주는 **Command** 방식이 있다.

Command 방식은 순간순간 필요한 옵션을 적어주기만 하면 되기 때문에 여기서는 **Interactive**
방식을 다루어 보기로 한다

Export 의 단위

Full 단위 : 전체 데이터베이스를 **Export** 한다.

User 단위 : 특정 유저 전체 오브젝트를 **Export** 한다.

Table 단위 : 특정 테이블을 **Export** 한다.

Partition 단위 : 특정 테이블의 파티션을 **Export** 한다.

Export 의 실제 예제

| 예 1 | 전체 데이터베이스의 **Export** (Interactive Method)

```
$ exp system/manager
Connected to: ORACLE8 Server Release 8.0.5 - Production
With the procedural and distributed options
PL/SQL Release 2.2 - Production
Enter array fetch buffer size : 4096 > 100000 (RETURN)
Export file : expdat.dmp >
(1) E(ntire database), (2) U(sers), (3) T(ables) : U > e
Export grants (Y/N) : Y > y
Export table data (Y/N) : Y > y
Compress extents (Y/N) : Y > y
About to export the entire database....
. exporting tablespace definitions
. exporting profiles
. exporting user definitions
. exporting role
. exporting rollback segment definitions
. exporting database links
. exporting sequence numbers
. exporting sequence numbers
. exporting cluster definitions
. exporting stored procedures
. about to export SYSTEM's tables ...
. about to export SCOTT's tables ...
. exporting synonyms
. exporting views
. exporting referential integrity constraints
. exporting triggers
```

```

Export terminated successfully without warnings.
| 예 2 | 전체데이터베이스의 EXPORT(Command Line Method)
$ exp userid=system/manager full=y file=fullbackup.dmp buffer=100000
| 예 3 | 전체데이터베이스의 EXPORT(Dynamic Method)
Export 패러미터를다음과같은파일(tusc.par) 형태로만든다.
system/manager
full=y
file=fullbackup.dmp
buffer=100000
$ exp parfile=tusc.par
| 예 4 | User 단위의 Export
$ exp system/manager
Connected to: ORACLE8 Server Release 8.0.5 - Production
With the procedural and distributed options
PL/SQL Release 2.2 - Production
Enter array fetch buffer size : 4096 > 100000 (RETURN)
Export file : expdat.dmp >
(1) E(ntire database), (2) U(sers), (3) T(ables) : U > u
Export grants (Y/N) : Y > y
Export table data (Y/N) : Y > y
Compress extents (Y/N) : Y > y
About to export specified users
User to be exported: (RETURN to quit) >scott
. exporting snapshots
. exporting snapshot log
. exporting database links
. exporting sequence numbers
. exporting sequence numbers
. exporting cluster definitions
. exporting stored procedures
. about to export SCOTT's tables ...
. exporting synonyms
. exporting views
. exporting referential integrity constraints
. exporting triggers
Export terminated successfully without warnings.
| 예 5 | User 단위의 Export (Command Line Method)
$ exp system/manager owner=scott file=scott.dmp buffer=100000
또는
$ exp scott/tiger file=scott.dmp buffer=100000
| 예 6 | User 단위의 Export (Dynamic Method )
Export 패러미터를다음과같은파일(tusc.par) 형태로만든다.
$vi tusc.par
scott/tiger
file=scott.dmp
buffer=1000000
$ exp parfile=tusc.par
| 예 7 | Table 단위의 Export
$ exp scott/tiger file=table.dmp tables=emp,dept buffer=100000

```

#덤프

>su - oracle

익스포트 > exp [계정명/비번] statistics=none

임포트 > imp [계정명/비번] statistics=safe

#에러시

EXP-00091 Exporting questionable statistics

1. exp 받는 동안 row error 가 있는 경우
2. client char set 이나 nchar char set 이 server char set 이나 nchar char set 과 맞지 않는 경우
3. exp 시 query 옵션이 사용된 경우
4. partitions, subpartitions 만 exp 받는 경우

위의 4 가지 경우인 경우 위의 warning 이 나타납니다.

위의 4 가지인 경우 기존의 통계정보와 달라질 수 있기 때문에 recalculate 가 필요합니다.

oracle 9 버전부터 export 시 default 로 기존의 통계정보를 export 받는데

export 받기 전 정상적으로 analyze 가 되지 않았다면 위에 해당하는 에러가 발생하게 됩니다.

answer1>

통계정보를 무시하고 export 할려고 한다면..옵션절에 **statistics = none** 로 써
주시고 export 받으면 됩니다.

answer2>

imp 받으실때 **statistics=safe** 로 받으면 됩니다.

여러 exp 파일이 있는 경우 저 warning 이 났는지 안났는지 모르기때문에 safe 옵션을 쓰면 재계산이 필요한 경우는 해주고 안필요하면 안하고 넘어가게 됩니다.

처음 imp 실패시 테이블을 생성해서 해당 테이블이 존재함으로 나오는 에러입니다.

imp 시 ignore=y 를 추가하면 되고요,

export 한 서버는 KO16MSWIN949 캐릭터 셋을 쓰고,

import 할 서버는 KO16KSC5601 이고 , NLS_LANG=XXXX.XXX.UTF8 로 설정하신 거 같네요

우선 import 할 서버의 NLS_LANG 값을 확인해 보시고요

export 한 서버는 캐릭터 셋을 봐서 윈도우의 기본 캐릭터 셋인데 혹시 윈도우인가요?

KO16MSWIN949 가 KO16KSC5601 의 super set 입니다.

KO16MSWIN949 가 KO16KSC5601 보다 더 많은 글자를 지원합니다.

import 시 KO16MSWIN949 에서 KO16KSC5601 로 캐릭터 셋 변환이 일어나는데,

캐릭터 셋 변환은 더 큰 집합으로 해야 문제가 없습니다.

KO16KSC5601 에서 지원하지 않는 글자가 있을 수 있기 때문입니다.

우선 NLS_LANG 를 KO16KSC5601 로 변환하시고, ignore=y 옵션을 추가하고 다시 imp 수행바랍니다.

그랬을 경우 문제가 발생하면 import 할 서버를 KO16MSWIN949 로 캐릭터셋을 변경하고 작업을 하셔야 합니다.

그럴 경우 NLS_LANG 도 같이 변경하시고요

테이블 전체 삭제

```
SQL> Begin
2   for c in (select table_name from user_tables) loop
3       execute immediate ('drop table '||c.table_name||' cascade constraints');
4   end loop;
5   End;
6   .
SQL> run
```

시퀀스 전체 삭제

```
SQL> Begin
2   for c in (select * from all_sequences where sequence_owner = 'DBUSER') loop
3       execute immediate 'drop sequence '||c.sequence_name;
4   end loop;
```

5 End;

6 .

SQL> run

테이블을 삭제하면 휴지통 테이블로 넘어가면서 복구가 가능한 상태로 변한다.

휴지통 테이블을 삭제한다.(BIN\$어쩌구....로 되어있다)

휴지통 테이블 확인 (select * from tab)으로도 확인 가능

SQL> SHOW RECYCLEBIN;

휴지통 테이블 삭제

SQL> PURGE RECYCLEBIN;

삭제 테이블 복구

SQL> FLASHBACK TABLE table_name PURGE;

[another] -----

* 테이블 삭제시 다음과 같은 에러를 냈을때 우회 방법

ORA-00701 object necessary for warmstarting database cannot be altered

Tool 안에서 스크립트 문구 만들기

table, index, sequence 등의 모든 object 를 삭제할때

SELECT 'drop '

||object_type

||' '

|| object_name

|| DECODE(OBJECT_TYPE,'TABLE',' CASCADE CONSTRAINTS;',';')

FROM user_objects

WHERE object_type in ('TABLE', 'SEQUENCE')