

ZÁLOHOVANIE DATABÁZY CEZ ORACLE DATAPUMP EXPORT

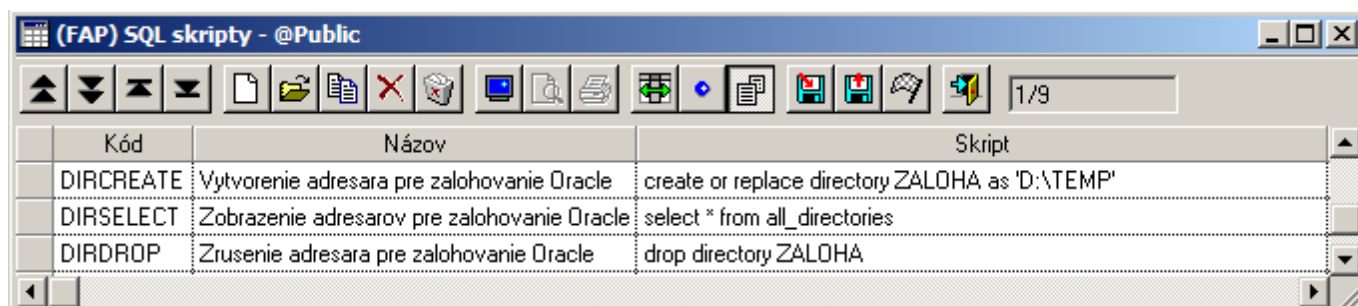
Export databázy vykonáva program Oracle Datapump (EXPDP.EXE), server pritom musí bežať, export je veľmi rýchly, výsledkom je jeden veľký binárny súbor. Tento súbor sa dá ešte celkom dobre zazipovať, napríklad cez Tangram Backup, pokiaľ nie je väčší ako 2GB. Restore sa robí programom IMPDP a dá sa obnoviť aj časť databázy (jedna schéma alebo jedna tabuľka). V záložnom adresári sa vytvára aj LOG file, pri každej zálohe sa prepisuje.

Zálohovanie vieme zatiaľ spustiť len priamo na serveri, kde je nainštalovaný aj program EXPDP.

Zálohovať sa dá len na lokálne disky (EXPDP asi nepodporuje sieťové disky), odtiaľ sa súbor zipuje cez Tangram Backup.

Nastavenia záložného adresára

Programy EXPDP a IMPDP musia mať v databáze vytvorený objekt typu DIRECTORY, ktorý obsahuje názov skutočného zálohovacieho adresára. Spustíte TangramEko (alebo Sklad), musíte sa prihlásiť ako vlastník schémy (alebo iný user s právami administrátora). Spustíte menu Systém – Nástroje – Skripty. Vytvorte tieto skripty:

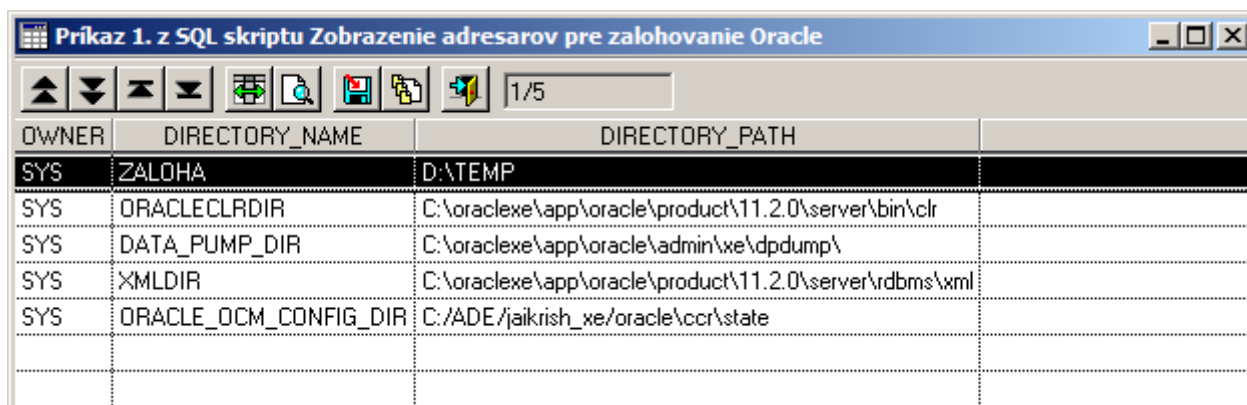


	Kód	Názov	Skript
	DIRCREATE	Vytvorenie adresara pre zalohovanie Oracle	create or replace directory ZALOHA as 'D:\TEMP'
	DIRSELECT	Zobrazenie adresarov pre zalohovanie Oracle	select * from all_directories
	DIRDROP	Zrusenie adresara pre zalohovanie Oracle	drop directory ZALOHA

DIRCREATE vytvorí v databáze odkaz na skutočný adresár D:\TEMP. Adresár musí byť vytvorený vo Windowse.

DIRDROP zruší z databázy odkaz na adresár. Skutočný adresár sa tým nevymaže.

DIRSELECT zobrazí aktuálny zoznam odkazov z databázy (obr. nižšie).



OWNER	DIRECTORY_NAME	DIRECTORY_PATH
SYS	ZALOHA	D:\TEMP
SYS	ORACLECLDIR	C:\oracle\app\oracle\product\11.2.0\server\bin\clr
SYS	DATA_PUMP_DIR	C:\oracle\app\oracle\admin\xe\dpdump\
SYS	XMLDIR	C:\oracle\app\oracle\product\11.2.0\server\rdbms\xml
SYS	ORACLE_OCM_CONFIG_DIR	C:/ADE/jaikrish_xe/oracle/ccr/state

Názov odkazu „ZALOHA“ (v stĺpci DIRECTORY_NAME) sa použije pri spustení EXPDP a IMPDP.

Príkazový súbor pre spustenie exportu

1. V adresári C:\ZALOHA vytvorte textový BAT súbor, napríklad ZALOHA.BAT, vložte do neho príkazy

```
DEL zaloha.dmp
```

```
EXPDP system/heslo directory=zaloha dumpfile=zaloha.dmp content=all schemas=data1,data2  
... alebo
```

```
EXPDP system/heslo directory=zaloha dumpfile=zaloha.dmp content=all full=yes
```

```
BACKUP.EXE auto
```

2. V riadku EXPDP zadajte meno a heslo usera, ktorý má pridelené práva na zálohovanie (prípadne administrátora)
3. Parameter DIRECTORY=ZALOHA obsahuje názov odkazu na adresár
4. Parameter DUMPFILE=ZALOHA.DMP obsahuje názov súboru, ktorý sa vytvorí v záložnom adresári
5. V riadku EXPDP zadajte zoznam schém (SCHEMAS), mali by sa zálohovať len pracovné schémy, nie archívne
6. Ak zvolíte FULL=YES, odzalohuje sa kompletná databáza so všetkými (aj systémovými) schémami
7. Riadok BACKUP.EXE vložte len vtedy, ak chcete vyexportovaný súbor ZALOHA.DMP zazipovať na ďalší disk. Pred spustením zálohovania nakonfigurujte program Tangram Backup (viď ďalej)

Nastavenie Tangram Backup

Program Backup.exe (Tangram Zip Backup) môžete zdarma stiahnuť zo stránky www.tangram.sk. Program sa neinštaluje, len ho skopírujte do adresára C:\ZALOHA.

Program nakonfigurujte podľa príkladu vpravo. Podrobný popis ovládania programu je v súbore Backup.chm (help súbor systému Windows). Tento príklad vezme súbor Zaloha.dmp z adresára C:\Zaloha a zazipuje ho na sieťový disk (\\Wd-netcenter\...). Udržiava sa posledných 5 záloh, staršie zálohy sa automaticky vymažú.

Automatické spúšťanie zálohy

Vytvorený súbor ZALOHA.BAT sa môže automaticky spúšťať pomocou štandardného plánovača v systéme Windows.

Ak nemáte radi Microsoft, môžete využiť náš vlastný plánovač Tangram Scheduler (spustiteľný aj ako service) alebo niektorý free plánovač (pyCron, nnCronLITE, ...).



OBNOVA DATABÁZY CEZ ORACLE DATAPUMP IMPORT

Obnovenie databázy vykonáva program Oracle Datapump Import (IMPDP.EXE), server pritom musí bežať. Obnoviť je možné naraz celú schému alebo jednotlivé tabuľky.

Príklad: Obnova schémy DEMO pod pôvodným názvom.

1. Ak v databáze existuje pôvodná schéma DEMO, najprv ju musíte vymazať:
Spustite **OracleManager**, otvorte „Zoznam schém“ a zrušte schému DEMO (ak existuje)
2. V databáze musí byť nakonfigurovaný záložný adresár (directory) podľa postupu v kapitole o exporte
3. Skopírujte príslušný záložný súbor (ZALOHA.DMP) do záložného adresára
4. Spustite príkazový riadok (konzolu) a zadajte príkaz:
IMPDP system/xxxxx directory=zaloha dumpfile=zaloha.dmp schemas=DEMO

Príklad: Obnova schémy DEMO pod novým názvom DEMONEW.

1. Ak v databáze existuje nová schéma DEMONEW, najprv ju musíte vymazať (viď vyššie)
2. V databáze musí byť nakonfigurovaný záložný adresár (directory) podľa postupu v kapitole o exporte
3. Skopírujte príslušný záložný súbor (ZALOHA.DMP) do záložného adresára
4. Spustite príkazový riadok (konzolu) a zadajte príkaz:
IMPDP system/xxxxx directory=zaloha dumpfile=zaloha.dmp remap_schema=DEMO:DEMONEW

Obnova jednotlivých tabuliek

1. Postup je rovnaký, ako pri obnove celej schémy. Ak tabuľka existuje, prepíše sa bez varovania.
2. Spustite príkazový riadok (konzolu) a zadajte príkaz:
IMPDP system/xxxxx DIRECTORY=zaloha DUMPFILE=zaloha.dmp TABLES=povodnaschema.tabu1ka REMAP_SCHEMA=povodnaschema:novaschema TABLE_EXISTS_ACTION=REPLACE
3. Parameter TABLES môže obsahovať viac tabuliek, oddelených čiarkami. Názov tabuľky musí obsahovať vlastníka (schému), ak je v zálohe viac schém naraz.
4. Parameter REMAP_SCHEMA nemusí byť, ak chceme obnoviť tabuľky do pôvodnej schémy

TABLESPACES + DATAFILES:

`CREATE TABLESPACE nazov DATAFILE 'C:\Oracle\Oradata\Xe\nazov.dbf' SIZE 100M AUTOEXTEND ON`
vytvorenie standardneho tablespace vo zvolenom adresari, default size 100MB, autoextend

`CREATE TEMPORARY TABLESPACE nazov`
vytvorenie standardneho tablespace, default size 100MB, autoextent

`DROP TABLESPACE nazov`
zrusenie tablespace (standard aj temporary)

`SELECT * FROM DBA_TABLESPACES`
`SELECT * FROM DBA_DATA_FILES`
zaznam o vsetkyh tablespaces (logicke oddiely) a datafiles (fyzicke subory) v databaze

`ALTER DATABASE DATAFILE 'C:\ORACLE\ORADATA\XE\SYSTEM.DBF' AUTOEXTEND ON MAXSIZE UNLIMITED`
nastavenie tablespace SYSTEM na neobmedzenu velkost (XE ma defaultne nastaveny 600M, ak je vela schem, nestaci to)

DOLEZITE SYSTEMOVE POHLADY (VIEWS)

`ALL_ALL_TABLES` - zoznam vsetkyh tabuliek v databaze
`ALL_OBJECTS` - zoznam vsetkyh objektov (tabulky, indexy, ...)
`ALL_TAB_COLUMNS` - zoznam vsetkyh stlpcov zo vsetkyh tabuliek
`ALL_TABLES` - zoznam vsetkyh tabuliek v databaze (cim sa lisi od `ALL_ALL_TABLES` ?)
`ALL_TAB_PRIVS` - pristupove prava
`ALL_USERS` - pouzivatelja (username + userid)
`DATABASE_PROPERTIES` - nastavenie databazy (NLS parametre a default hodnoty)
`NLS_DATABASE_PARAMETERS` - parametre databazy
`NLS_INSTANCE_PARAMETERS` - parametre databazy
`NLS_SESSION_PARAMETERS` - parametre databazy
`ROLE_ROLE_PRIVS` - role pre rolu
`ROLE_SYS_PRIVS` - systemove prava pre rolu
`ROLE_TAB_PRIVS` - objektove prava pre rolu
`SESSION_CONTEXT`, `SESSION_PRIVS`, `SESSION_ROLES` - pristupove prava
`SYSFILES` - databazove subory (podobne ako `DBA_DATA_FILES`)
`TABLE_PRIVILEGES` - pristupove prava
`SYS.V_$VERSION` - verzie jednotlivych komponentov Oracle
`SYS.V_$SESSION` - beziace procesy:
`select type, program, username, schemaname, sid from SYS.V_$session order by sid;`

PASSWORDS & PROFILES

Nastavenie default profilu – neobmedzená expirácia hesiel:

`alter profile default limit PASSWORD_LIFE_TIME UNLIMITED;`
`alter profile default limit PASSWORD_LOCK_TIME UNLIMITED;`
`alter profile default limit PASSWORD_GRACE_TIME UNLIMITED;`
`alter profile default limit FAILED_LOGIN_ATTEMPTS počet|UNLIMITED`

Zmena profilu sa prejaví buď pri vytvorení nového usera, alebo po priradení profilu userovi:

`ALTER USER meno PROFILE default;`

Zistenie aktualneho nastavenia profilov a hesiel:

`SELECT * FROM DBA_PROFILES;`
`SELECT * FROM DBA_USERS;`

AUDIT LOG

`NOAUDIT ALL;`
`DELETE FROM SYS.AUD$;`

TEMPORARY TABULKY

User s minimálnymi pravami CREATE SESSION a CREATE TABLE moze vytvorit docasnu tabulku, k nej docasny index a potom to vsetko zrusit. Ak by sme nedali GLOBAL TEMPORARY, tak sa pokusa vytvorit tabulku v default tablespace, na ktory nema pridelenu kvotu alebo nejake prava a vyhodi chybu. Ak zadame G.T., tak sa tabulka vytvori automaticky v TEMP tablespace ku ktorému majú všetci prístup. Ak neurobíme DROP, hlavička tabulky zostane ale riadky sa automaticky dropnu (to je vlastnosť GT tabuliek). Ak sa nezadá ON COMMIT ... tak sa riadky dropnu pri prvom commit, inak sa dropnu pri ukončení session. Viaceri používatelia si môžu vytvoriť tabulku s rovnakým názvom, každý si ju vytvorí vo svojej schéme. Právo CREATE TABLE umožňuje aj vytvoriť index a dropnúť svoju vlastnú tabulku. Správne by sme na Oracle mali vytvoriť GLOBAL TEMPORARY tabulky už v návrhu dátového modelu. Používatelia potom ani nepotrebujú právo CREATE TABLE, každý user môže začať INSERT do tabulky, každému sa vytvorí jeho vlastná kópia tela tabulky, riadky sa automaticky dropnu na konci transakcie alebo session.

```
CREATE GLOBAL TEMPORARY TABLE nazov (stlpce) {ON COMMIT PRESERVE ROWS}
CREATE INDEX nazov_indexu ON nazov_tabulky(stlpce)
DROP TABLE nazov
```

ZMENA KÓDOVEJ STRÁNKY Oracle/XE NA WIN-1250

pripojíš sa ako SYSDBA v sqlplus

```
SHUTDOWN IMMEDIATE;
STARTUP MOUNT;
ALTER SYSTEM ENABLE RESTRICTED SESSION;
ALTER SYSTEM SET JOB_QUEUE_PROCESSES=0;
ALTER SYSTEM SET AQ_TM_PROCESSES=0;
ALTER DATABASE OPEN;
ALTER DATABASE CHARACTER SET INTERNAL_USE EE8MSWIN1250;
SHUTDOWN;
STARTUP RESTRICT;
SHUTDOWN;
STARTUP;
```

ŠTART / STOP ORACLE DATABÁZY

Oracle database server je spustený ako servis. Štart a stop sa dá vykonať z príkazového riadka:

```
net start OracleXETNSListener
net start OracleServiceXE
net stop OracleServiceXE
net stop OracleXETNSListener
```

Databáza je obsluhovaná dvoma servismi (OracleService a OracleListener), obidva sa musia spustiť a zastaviť. V názve servisov je tzv SID databázy (v našom prípade „XE“ pre Oracle Express Edition). Ak neviete SID databázy, pozrite v systéme do zoznamu Services, všetky služby začínajú „Oracle...“.

ZMENA POČTU PROCESOV, SESSIONS, VEĽKOSTI SÚBOROV

OracleXE má štandardne nastavený maximálny počet procesov na 40 a niekedy to nestačí (prejaví sa to tak, že sa k databáze nedá pripojiť a musí sa reštartovať). Zmena sa prejaví až po reštarte databázy. Nastavenú hodnotu parametra PROCESSES môžeme zistiť (pre XE) cez web manažera Administration - About Database - Parameters.

Pripojiť sa ako SYSDBA v SQLPLUS a napísať jediný príkaz:

```
ALTER SYSTEM SET PROCESSES=120 SCOPE=SPFILE;
```

Môžeme tiež zrušiť obmedzenie počtu sessions (na XE štandardne 20):

```
ALTER SYSTEM RESET SESSIONS SCOPE=SPFILE SID='*'
```

Zrušenie obmedzenia na veľkosť datafile, ak to niekto nenastaví už pri vytvorení súboru.

```
alter database datafile 'd:\oraclexe\oradata\XE\USERS.DBF' autoextend on maxsize unlimited;
```

EDITOVANIE SPFILE

Spfile je binárny súbor a nedá sa editovať priamo. Dá sa skonvertovať na text, editovať a potom vrátiť späť na binárny:

```
STARTUP NOMOUNT the instance
CREATE PFILE FROM SPFILE;
```

Using a text editor, modify the pfile. The pfile will be in the default location.

```
CREATE SPFILE FROM PFILE;
ALTER DATABASE OPEN;
```

Toto by mal byť rýchlejší postup (po ALTER musíme reštartovať databázu, aby sa zmena prejavila).

```
ALTER SYSTEM SET PROCESSES=120 SCOPE=SPFILE;
```

(pre XE je SPFILE v adresári: c:\oraclexe\app\oracle\product\10.2.0\server\dfs\SPFILEXE.ORA)

POŠKODENÁ DATABÁZA

Ak sa počítač s databázou násilne vypne, môže sa stať, že po zapnutí databáza nenabehne.

1. Ak databáza nie je poškodená, stačí vypnúť a znova zapnúť PC (musí prebehnúť štandardný Windows shutdown)
2. Ak to nepomôže, spustíte SQLPLUS, prihláste sa ako "SYS AS SYSDBA" a skúste tzv. General Recovery:

```
SHUTDOWN;                (ukončiť nesprávne naštartovanú databázu)
STARTUP MOUNT;            (odpoveď "database mounted" je OK, inak je zle)
ALTER DATABASE RECOVER;   (odpoveď "database altered" je OK, inak je zle)
SHUTDOWN;                (reštart databázy, môže sa reštartovať Windows)
STARTUP;
```

Ak je poškodený napríklad SYSTEM.DBF, tak skúsiť:

```
RECOVER DATABASE USING BACKUP CONTROLFILE;
... mal by si vyžiadať LOG-file, podarilo sa nám vložiť názov najstaršieho LOG z adresára
C:\Oracle\App\Oracle\Fast_recovery_area\XE\Onlineolog\*.LOG
odpoveď „Media recovery done“ je OK
ALTER DATABASE OPEN RESETLOGS;
```

Ak je poškodený LOG file ("open failed for members of log group x of thread y")

```
ALTER DATABASE CLEAR LOGFILE GROUP x
```

UŽITOČNÉ SQL PRÍKAZY

Náhrada za IN / NOT IN (SELECT ...), ktorý je v Oracle veľmi pomalý:

```
NOT EXISTS (SELECT NULL FROM TOVARY WHERE XX=D.KOD)
```

```
SELECT * FROM PCKasadokh WHERE EXISTS (SELECT null FROM PCKasadokr WHERE
(PCKasadokh.xx=PCKasadokr.xh) and (PCKasadokr.zmazany is not null))
```

Hromadný update jednej tabulky podľa hodnôt v druhej tabulke (jedna položka):

```
update TOVARY set povsort=(select povsort from DRUHA where TOVARY.xx=DRUHA.xx);
```

Tabuľka DRUHA obsahuje stĺpce XX (kod tovaru) a POVSORT (hodnota, ktorú chcem vložiť do TOVARY).

Hromadný update jednej tabulky podľa hodnôt v druhej tabulke (tri položky naraz):

```
update TOVARY set (povsort,klasifikacia,colsadz)=(select povsort,klasifikacia,colsadz from
DRUHA where TOVARY.xx=DRUHA.xx);
```

Cyklus v SQL skripte (SQL Developer). Lomítko na konci musí byť, inak nenájde koniec bloku !

```
-- ANONYMNA PROCEDURA
DECLARE
    i number;
BEGIN
    FOR i IN 1..299 LOOP
        insert into c_str (xx, nazov)
        values ('STR' || to_char(i,'000'), 'Stredisko generované ' || to_char(i,'000'));
    END LOOP;
END;
/
```