



ORACLE®  
DATABASE

# Agenda

1. Export / Import
2. Datapump
3. Datapump Parameter
4. Übungen

# Export / Import Übersicht

- Mit Export / Import Tools können ganze Datenbanken- oder Teile davon exportiert- und in anderen Datenbanken wieder importiert werden
- Bei ORACLE DB's gibt es grundsätzlich ein Tool um Daten zu exportieren aber mehrere Tools um Daten zu importieren bzw. zu laden
  - Hier wird nur das klassische Import Tool betrachtet

- Früher wurden gerne Fullexporte zu Backupzwecken gemacht
  - Heutzutage werden die Datenbanken immer grösser, Fullexporte als Backup's sind nicht mehr praktikabel
- Kopieren von einzelnen Schemen- oder Teilen davon
- “saubere” Upgrades von Datenbanken ohne alte Zöpfe von früheren Migrationen mitzuschleppen
  - «Full Imports» möglichst vermeiden, nur die Schemen importieren, die benötigt werden
- Weitergeben von Teilen der Datenbank an Drittpersonen wie z.B. SW Entwicklungsfirmen
  - oder umgekehrt 😊
- Reorganisieren von Tabellen

- Anders als bei RMAN erwartet das Import Tool eine laufende Datenbank
- Eine laufende DB bedeutet, dass mindestens alle sys Objekte vorhanden sind
- **→ sys Objekte werden nicht exportiert und können auch nicht importiert werden**
- Dies ist konzeptionell korrekt, muss aber berücksichtigt werden
- Erstellt man keine Objekte unter dem Schema sys, was man eh nicht machen sollte, ist das kein Problem

- Bei ORACLE gibt es alte Export / Import Utilities, welche nicht mehr verwendet werden sollten
  - exp/imp
  - Werden “Export” und “Import” genannt
  - imp steht aus Kompatibilitätsgründen weiterhin zur Verfügung damit alte Dumps importiert werden können
  - exp ist deprecated, steht jedoch in Version 12.1 noch zur Verfügung
- Die neuen Export / Import Utilities werden «Datapump» genannt
  - expdp / impdp
  - Datapump wurde mit ORACLE 10g eingeführt

# Agenda

1. Export / Import
2. Datapump
3. Datapump Parameter
4. Übungen

- Datapump steht als CLI Tools (expdp / impdp) oder auch als PL/SQL API (dbms\_datapump) innerhalb der Datenbank zur Verfügung
- Wird ein Datapump Job gestartet, läuft er via dbms\_datapump innerhalb der Datenbank. Das CLI Interface kann grundsätzlich gestoppt werden ohne dass der DP Job gestoppt wird
- Es ist möglich innerhalb der Datenbank via dbms\_datapump Schemen aus anderen Datenbanken in die aktuelle Datenbank zu importieren
  - z.B. als eine Art der Replikation oder zum Aktualisieren von Schemen in Entwicklungs- oder Test Datenbanken
- Wir beschränken uns hier auf die CLI Tools. Das PL/SQL API dbms\_datapump ist in der ORACLE Dokumentation beschrieben
  - [http://docs.oracle.com/database/121/ARPLS/d\\_datpmp.htm#ARPLS356](http://docs.oracle.com/database/121/ARPLS/d_datpmp.htm#ARPLS356)



- Da Datapump innerhalb der Datenbank läuft, werden die Dump Files aus der Datenbank in das Filesystem des DB Servers geschrieben
  - Auch wenn der DP Job remote auf einem Client gestartet wird
  - Bei exp wurden die Dumpfiles immer dorthin geschrieben, wo exp gestartet wurde
- Wo die Files ins Filesystem geschrieben werden wird über «Directories» gesteuert
  - Siehe Folien 20, Objekte
- DD View:

**DBA\_DIRECTORIES**

- Das create statement macht vieles klar ☺

```
create or replace directory datapump  
as '/u00/app/oracle/admin/DAH11203/dpdump' ;
```

- ORACLE: Eine Firma, viele Entwickler 😊
- Leider ist die Philosophie bei den CLI Tools von ORACLE nicht einheitlich...
  - ...wenigstens ist exp und expdp bzw. imp und impdp von der Philosophie her gleich 😊
- Wie immer ist es wichtig zu wissen wie die Hilfe aufgerufen wird
  - Bei sehr vielen UNIX- oder ORACLE Produkten wird die Hilfe mit den Parametern «-h» oder «--help» in verschiedenen Ausprägungen aufgerufen
  - Nicht so bei expdp/impdp 😞

```
expdp help=yes  
impdp help=yes
```

- Bei Datapump CLI Befehlen können die Parameter wie üblich auf der Kommandozeile mit angegeben werden
  - Das sieht dann etwa so aus:

```
expdp USERID=system/manager FULL=Y PARALLEL=8  
FLASHBACK_TIME=sysdate JOB_NAME=dp_fullexp  
DIRECTORY=DATA_PUMP_DIR DUMPFILE=dp_fullexp_%U.dmp  
LOGFILE=dp_fullexp.log
```

- → etwas unübersichtlich und schwer nachvollziehbar
- Exporte und Importe müssen oft wiederholt werden z.B. zum Aktualisieren von Schemen oder Tabellen oder weil der Import oder Export nicht ganz so geklappt hat wie erwartet weil etwas vergessen wurde

- Die Parameter können auch in einem Parameterfile angegeben werden
- Dies ist übersichtlicher und es kann nachvollzogen werden wie der Export / Import das letzte Mal gemacht wurde

```
USERID="/ as sysdba"  
FULL=Y  
FLASHBACK_TIME=sysdate  
JOB_NAME=dp_fullexp1  
DIRECTORY=dpdump  
DUMPFILE=dp_fullexp_%U.dmp  
LOGFILE=dp_fullexp.log
```

- Start von expdp mit Angabe eines Parameterfiles

```
expdp parfile=exp.par
```

# Agenda

1. Export / Import
2. Datapump
3. Datapump Parameter
4. Übungen

- Es stehen einige Parameter für den Export zur Verfügung
- Man muss nicht alle Parameter kennen, wie immer muss man wissen wo man sie nachschauen kann 😊
- Es ist wichtig, dass man eine grobe Idee hat was gemacht werden kann, Details dazu können dann in der Dokumentation nachgeschaut werden
- Anschliessend werden ein paar Parameter vorgestellt. Diese Liste ist nicht vollständig.
- Datapump ist im «Utilities Guide» dokumentiert
- [http://http://docs.oracle.com/cd/E11882\\_01/server.112/e22490/toc.htm](http://http://docs.oracle.com/cd/E11882_01/server.112/e22490/toc.htm)

## Part I Oracle Data Pump

- ▶ 1 Overview of Oracle Data Pump
- ▶ 2 Data Pump Export
- ▶ 3 Data Pump Import
- ▶ 4 Data Pump Legacy Mode
- ▶ 5 Data Pump Performance
- ▶ 6 The Data Pump API

# Datapump Export Parameter 1/3

- Die vollständige Liste der Parameter ist in der Dokumentation ersichtlich
- Es lohnt sich die Liste einmal durchzuschauen um eine Idee der Möglichkeiten zu bekommen
- Was ich oft als Basis verwende:

```
USERID="/ as sysdba"  
FULL=Y  
PARALLEL=4  
FLASHBACK_TIME=sysdate  
JOB_NAME=dp_fullexp  
DIRECTORY=dpdump  
DUMPFILE=dp_fullexp_%U.dmp  
LOGFILE=dp_fullexp.log
```

# Datapump Export Parameter 2/3

Parameter	Bedeutung	Beispiel
USERID	User mit welchem man sich in die DB verbindet	[Username]/[Passwort]@tns_name system/manager "/ as sysdba" system/manager@xe112.tsbe.ch
FULL	Fullexport der gesamten DB Wird "N" angegeben, muss eine sonstige Option wie z.B. ein Schema oder eine Tabelle angegeben werden	Y oder N
FLASHBACK_ TIME	Garantiert einen konsistenten Export. Wichtiger Parameter!	sysdate
DIRECTORY	Name des DB Directories, wohin der Export gemacht werden soll	Name aus dba_directories
PARALLEL	Parallelität	8
DUMPFIL	Name des Dumpfiles. Die Variable %U gibt an, dass mehrere Files erstellt werden können. z.B. wenn Parallel > 1 eingestellt ist	Beliebiger Name Ggf. Mit %U



# Datapump Export Parameter 3/3

- Was soll exportiert werden (FULL=N)

Parameter	Bedeutung	Beispiel
SCHEMAS	Komma separierte Liste der Schemas, die exportiert werden sollen	hr,scott,webshop webshop
TABLES	Komma separierte Liste der Tabellen, die exportiert werden sollen	webshop.kunden
TABLESPACES	Komma separierte Liste der Tablespaces, die exportiert werden sollen	USERS,WEBSTOP,TOOLS
EXCLUDE	Exkludieren von Tabellen	TABLE:"IN ('COUNTRIES', 'LOCATIONS', 'REGIONS')"
QUERY	Einschränkung auf Rows, die exportiert werden sollen	employees:"WHERE department_id !=20 ORDER BY employee_id"

# Datapump Import Parameter 1/4

- Die vollständige Liste der Parameter ist in der Dokumentation ersichtlich
- Es lohnt sich die Liste einmal durchzuschauen um eine Idee der Möglichkeiten zu bekommen
- Was ich oft als Basis verwende:

```
USERID="/ as sysdba"  
FULL=N  
PARALLEL=8  
JOB_NAME=db_import  
DIRECTORY=dpump_dir  
SCHEMAS=hr  
EXCLUDE=statistics  
DUMPFILE=dp_fullexp_%U.dmp  
LOGFILE=dp_imp_hr.log
```

# Datapump Import Parameter 2/4

- Parameter FULL:
  - FULL=Y heisst, dass alles importiert wird, was sich im Dumpfile befindet. Das kann ein Full Export sein, muss aber nicht.
  - Beim Import sollte wenn möglich mit FULL=N gearbeitet werden
  - FULL=Y importiert bei einem Fullexport eine ganze Datenbank und generiert sehr viele Fehler wie uA Constraintverletzungen
    - Das Kontrollieren des Logfiles ist sehr aufwändig!
  - FULL=Y importiert allen alten «Mist», den man meist gar nicht mehr braucht
  - Default ist FULL=Y

# Datapump Import Parameter 3/4

Parameter	Bedeutung	Beispiel
USERID	User mit welchem man sich in die DB verbindet	[Username]/[Passwort]@tns_name system/manager "/ as sysdba" system/manager@xe112.tsbe.ch
FULL	Fullimport der gesamten DB Wird "N" angegeben, muss eine sonstige Option wie z.B. ein Schema oder eine Tabelle angegeben werden	Y oder N
DIRECTORY	Name des DB Directories, worin sich die Dump Files befinden	Name aus dba_directories
PARALLEL	Parallelität	8
DUMPFIL	Name des Dumpfiles. Die Variable %U gibt an, dass mehrere Files erstellt werden können. z.B. wenn Parallel > 1 eingestellt ist	Beliebiger Name Ggf. Mit %U
EXCLUDE	Was nicht importiert werden soll. Statistiken sollten neu generiert werden. Das Importieren dauert lange.	STATISTICS INDEX:"LIKE 'EMP%' " PROCEDURE

# Datapump Import Parameter 4/4

Parameter	Bedeutung	Beispiel
CONTENT	Was soll importiert werden	[ALL   DATA_ONLY   METADATA_ONLY]
SCHEMAS	Komma separierte Liste der Schemas, die importiert werden sollen	hr,scott,webshop webshop
TABLES	Komma separierte Liste der Tabellen, die importiert werden sollen	webshop.kunden
TABLESPACES	Komma separierte Liste der Tablespaces, die importiert werden sollen	USERS,WEBSHOP,TOOLS
QUERY	Einschränkung auf Rows, die importiert werden sollen	employees:"WHERE department_id !=20 ORDER BY employee_id"
REMAP_SCHEMA	Änderung des Schemaowners Parameter kann mehrfach vorkommen	[source_schema]: [dest_schema] webshop:webshop_copy
REMAP_TABLESPACE	Änderung des Tablespaces Parameter kann mehrfach vorkommen	[source_tablespace]: [dest_tablespace] USERS:USERS_NEU

# Agenda

1. Export / Import
2. Datapump
3. Datapump Parameter
4. Übungen

# Datapump und ORACLE XE 11gR2 1/2

- ORACLE XE ist die Gratisversion von ORACLE mit einigen Einschränkungen
  - z.B. können keine Patches zur Fehlerbehebung eingespielt werden
- Lancieren ORACLE neue Features, werden die üblicherweise forciert, damit sie auch eingesetzt werden (und «verbessert» werden können 😊)
- Bei XE 10gR2 hat Datapump funktioniert
- Bei XE 11gR2 funktioniert Datapump nicht 😞

```
oracle@vm100:/u01/app/oracle/admin/XE112/dmp/ [XE112] expdp parfile=expdp.par

Export: Release 11.2.0.2.0 - Production on Thu Jun 2 21:29:27 2016

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.

Connected to: Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit
Production
ORA-39006: internal error
ORA-39213: Metadata processing is not available
```

- ORACLE hat bei XE 11gR2 ein Directory im ORACLE\_HOME «vergessen»...
- Abhilfe: Hinkopieren des fehlenden Directories...

```
cp -p /tmp/xsl.tgz $ORACLE_HOME/rdbms/xml/  
cd $ORACLE_HOME/rdbms/xml/  
tar -xzf xsl.tgz  
rm xsl.tgz
```

- ...und neu Laden der Stylesheets in die Datenbank
  - Als User sys ☺

```
SQL> execute dbms_metadata_util.load_stylesheets
```



# Übung 1: Erstellen eines Full Exports

- Erstellen eines Full Exports der Datenbank
- Der Export soll in das Directory \$ORACLE\_BASE/admin/\$ORACLE\_SID/dmp gemacht werden
- 1x Full Export mit CLI Parametern
- 1x Full Export mit Parameterfile

# Übung 2: Import

- Importieren des Schemas webshop in die gleiche Datenbank unter dem Namen webshop\_copy
- Wahlweise mit Parameterfile oder direct via CLI

# Übung 3: Kontrolle Import

- Wurde das Schema webshop tatsächlich unter dem User webshop\_copy neu erstellt?
- In welchem Tablespace liegen die Tabellen?
- Wurde die View auch importiert?

# Übung 4: Import webshop 2

- Die Tabellen müssen in einem eigenen Tablespace liegen.
- Schema webshop\_copy inkl. aller dazugehörenden Objekte löschen
- Schema neu importieren (wurde mit Parameterfile gearbeitet, das jetzt wiederverwendet werden könnte? 😊) und darauf achten, dass die Tabellen im Tablespace webshop\_copy\_data landen.
- Kontrolle ob die Tabellen im richtigen Tablespace liegen

# Lösung Übung 1: Full Export 1/2

- Erstellen eines Full Exports der Datenbank
- Der Export soll in das Directory \$ORACLE\_BASE/admin/\$ORACLE\_SID/dmp gemacht werden
- 1x Full Export mit CLI Parametern
- 1x Full Export mit Parameterfile

```
SQL> create or replace directory dpdump as '/u01/app/  
oracle/admin/XE112/dmp' ;
```

```
expdp USERID=system/manager FULL=Y  
FLASHBACK_TIME=sysdate DIRECTORY=dpdump  
DUMPFILE=dp_fullexp.dmp LOGFILE=dp_fullexp.log
```

# Lösung Übung 1: Full Export 2/2

- Erstellen eines Full Exports der Datenbank
- Der Export soll in das Directory \$ORACLE\_BASE/admin/\$ORACLE\_SID/dmp gemacht werden
- 1x Full Export mit CLI Parametern
- 1x Full Export mit Parameterfile

**Parameterfile:**

**USERID="/ as sysdba"**

**FULL=Y**

**FLASHBACK\_TIME=sysdate**

**JOB\_NAME=dp\_fullexp**

**DIRECTORY=dpdump**

**DUMPFIL=dp\_fullexp.dmp**

**LOGFILE=dp\_fullexp.log**

**expdp parfile=expdp.par**

# Lösung Übung 2: Import

- Importieren des Schemas webshop in die gleiche Datenbank unter dem Namen webshop\_copy
- Wahlweise mit Parameterfile oder direct via CLI

```
Parameterfile:  
USERID="/ as sysdba"  
FULL=N  
PARALLEL=1  
JOB_NAME=db_imp_webshop  
DIRECTORY=dpdump  
REMAP_SCHEMA=webshop:webshop_copy  
SCHEMAS=WEBSHOP  
EXCLUDE=statistics  
DUMPFILE=dp_fullexp.dmp  
LOGFILE=dp_imp_webshop1.log  
  
impdp parfile=imp.par
```

# Lösung Übung 3: Kontrolle Import

- Wurde das Schema webshop tatsächlich unter dem User webshop\_copy neu erstellt?
- In welchem Tablespace liegen die Tabellen?
- Wurde die View auch importiert?

```
SQL> select table_name from dba_tables where owner =  
'WEBSHOP_COPY';
```

```
SQL> select table_name, tablespace_name from dba_tables  
where owner = 'WEBSHOP_COPY';
```

```
SQL> select owner, view_name from dba_views where owner  
like 'WEBSHOP%';
```



# Lösung Übung 4: Import webshop 2 1/2

- Die Tabellen müssen in einem eigenen Tablespace liegen.
- Schema webshop\_copy inkl. aller dazugehörenden Objekte löschen
- Schema neu importieren (wurde mit Parameterfile gearbeitet, das jetzt wiederverwendet werden könnte? ☺) und darauf achten, dass die Tabellen im Tablespace webshop\_copy\_data landen.
- Kontrolle ob die Tabellen im richtigen Tablespace liegen

```
SQL> drop user webshop_copy cascade;
```

Wo soll das Datenfile des Tablespaces WEBSHOP\_COPY\_DATA liegen und wie gross soll es werden?

```
SQL> select file_name, bytes/1024/1024 from  
dba_data_files where tablespace_name = (select distinct  
tablespace_name from dba_tables where owner =  
'WEBSHOP');
```

Tablespace erstellen

```
SQL> create tablespace webshop_copy_data datafile '/  
u02/oradata/XE112/webshop_copy_data01XE112.dbf' size  
50M;
```

# Lösung Übung 4: Import webshop 2 2/2

- Die Tabellen müssen in einem eigenen Tablespace liegen.
- Schema webshop\_copy inkl. aller dazugehörenden Objekte löschen
- Schema neu importieren (wurde mit Parameterfile gearbeitet, das jetzt wiederverwendet werden könnte? ☺) und darauf achten, dass die Tabellen im Tablespace webshop\_copy\_data landen.
- Kontrolle ob die Tabellen im richtigen Tablespace liegen

## **Parameterfile:**

```
USERID="/ as sysdba"
```

```
FULL=N
```

```
DIRECTORY=dpdump
```

```
REMAP_SCHEMA=webshop:webshop_copy
```

```
REMAP_TABLESPACE=WEBSHOP_DATA:WEBSHOP_COPY_DATA
```

```
SCHEMAS=WEBSHOP
```

```
EXCLUDE=statistics
```

```
DUMPFIL=dp_fullexp.dmp
```

```
LOGFILE=dp_imp_webshop2.log
```

```
impdp parfile=imp_webshop2.par
```

```
SQL> select table_name, tablespace_name from dba_tables  
where owner = 'WEBSHOP_COPY';
```

# Fragen?

