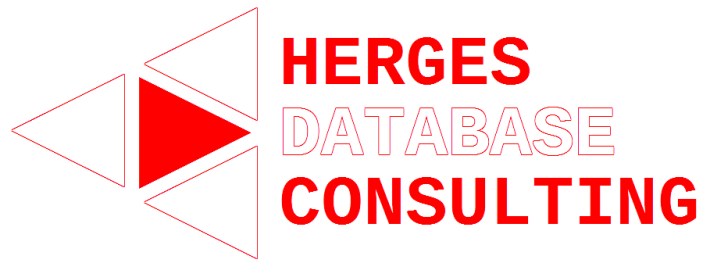

RMAN Reloaded

Recovery Manager **12^c**
News & Features



Oliver Herges

Oracle Database Principal Consultant
Freelancer

Oliver.Herges@gmx.de

www.xing.com/profile/Oliver_Herges

Oliver Herges – Database Experience

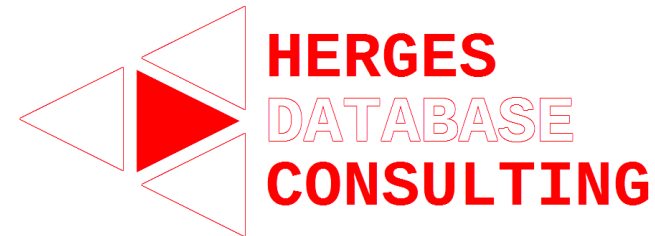
```
SELECT JOB FROM EMP WHERE ENAME = 'HERGES';
```

▶ Oracle Technical Consulting / DBA

- Über 16 Jahre Erfahrung in Konzeption & Implementierung von Oracle Datenbank-Infrastrukturen
- Gesamter Produkt- und Projekt-Lebenszyklus
- Projektleitung

▶ Lieblingsprodukte

- Oracle Database Server / Core-Technology
- Oracle Real Application Clusters (RAC)
- Oracle DataGuard
- Oracle Fail Safe
- Oracle Clusterware / Infrastructure
- Oracle Recovery Manager
- Oracle Exadata Database Machine



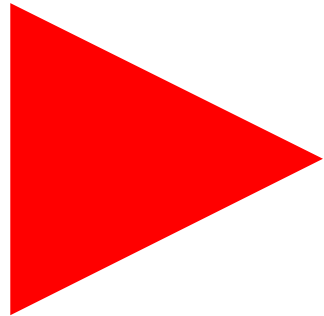
Agenda

RMAN Reloaded 12c

- ▶ Multitenant-Architektur & RMAN
 - Container Database
 - Pluggable Databases

 - ▶ RMAN 'Classic Improvements': "11g meets 12c"
 - Multisection-Backup Improvements
 - Active Database Duplication Improvements
 - Duplicate Improvements
 - Restoring & Recovering Files Over a Network
 - Cross-Platform Backup & Restore Improvements
 - Recovering Tables & Table Partitions from RMAN Backups

 - ▶ More Features!
-



Multitenant & RMAN

ORACLE®
DATABASE **12^c**

Multitenant

"PLUGGABLE DATABASE"

- ▶ RMAN-Integration für Backup & Recovery von CDBs und PDBs
 - Gesamte **Container-Datenbank (CDB)** mit CDB\$ROOT, PDB\$SEED und PDBs
 - Eine oder mehrere **Pluggable-Datenbanken (PDBs)**
 - Vorteil: Simplifizierung & Konsolidierung des Backups
- ▶ RMAN-Kommandos "**BACKUP | RESTORE | RECOVER ... DATABASE**" werden im Kontext der Datenbank anmeldung interpretiert und ausgeführt
 - Anmeldung an **CDB (Root-Container)**: Sicherung des gesamten Containers mit PDBs
 - Anmeldung an **PDB**: Sicherung der PDB
- ▶ Erweiterte Syntax: "**PLUGGABLE DATABASE**"
 - Zusätzliche Ebene für das Handling und die Sicherung & Wiederherstellung von PDBs
 - Beispiel:

```
% rman target sys/<password>@cdb  
RMAN> BACKUP PLUGGABLE DATABASE pdb1, pdb2;
```

Multitenant

"REPORT SCHEMA"

▶ RMAN> REPORT SCHEMA;

▶ List of Permanent Datafiles

=====

File	Size(MB)	Tablespace	RB	segs	Datafile Name
1	790	SYSTEM	YES		D:\ORADATA\CDB\SYSTEM01.DBF
2	250	PDB\$SEED:SYSTEM	NO		D:\ORADATA\CDB\PDBSEED\SYSTEM01.DBF
3	670	SYSAUX	NO		D:\ORADATA\CDB\SYSAUX01.DBF
4	505	PDB\$SEED:SYSAUX	NO		D:\ORADATA\CDB\PDBSEED\SYSAUX01.DBF
5	990	UNDOTBS1	YES		D:\ORADATA\CDB\UNDOTBS01.DBF
6	5	USERS	NO		D:\ORADATA\CDB\USERS01.DBF
7	260	PDB1:SYSTEM	NO		D:\ORADATA\CDB\PDB1\SYSTEM01.DBF
8	535	PDB1:SYSAUX	NO		D:\ORADATA\CDB\PDB1\SYSAUX01.DBF
9	5	PDB1:USERS	NO		D:\ORADATA\CDB\PDB1\PDB1_USERS01.DBF
10	260	PDB2:SYSTEM	NO		D:\ORADATA\CDB\PDB2\SYSTEM01.DBF
11	535	PDB2:SYSAUX	NO		D:\ORADATA\CDB\PDB2\SYSAUX01.DBF
12	5	PDB2:USERS	NO		D:\ORADATA\CDB\PDB2\PDB2_USERS01.DBF

List of Temporary Files

=====

File	Size(MB)	Tablespace	Maxsize(MB)	Tempfile Name
1	66	TEMP	32767	D:\ORADATA\CDB\TEMP01.DBF
2	61	PDB\$SEED:TEMP	32767	D:\ORADATA\CDB\PDBSEED\TEMP01.DBF
3	20	PDB1:TEMP	32767	D:\ORADATA\CDB\PDB1\TEMP01.DBF
4	20	PDB2:TEMP	32767	D:\ORADATA\CDB\PDB2\TEMP01.DBF

Multitenant

"BACKUP..." - Beispiele mit Container-Datenbank

- ▶ Anmeldung an der CDBs:

```
▪ % rman target sys/<password>@cdb
```

- ▶ Vollständige Sicherung der CDB, inklusive CDB\$ROOT, PDB\$SEED und allen PDBs:

```
▪ RMAN> BACKUP DATABASE PLUS ARCHIVELOG DELETE ALL INPUT;
```

- ▶ Sicherung von einzelnen PDBs:

```
▪ RMAN> BACKUP PLUGGABLE DATABASE pdb1, pdb2;
```

- ▶ Sicherung des Root-Containers:

```
▪ RMAN> BACKUP DATABASE ROOT;
```

Multitenant

"BACKUP..." - Beispiele mit Container-Datenbank

- ▶ Anmeldung an der CDBs:

```
▪ % rman target sys/<password>@cdb
```

- ▶ Sicherung von einzelnen Tablespaces über das Prefix der PDB:

```
▪ RMAN> BACKUP TABLESPACE PDB1:SYSTEM, PDB1:SYSAUX;
```

- ▶ Sicherung eines Tablespaces der Root-CDB:

```
▪ RMAN> BACKUP TABLESPACE SYSAUX;
```

- ▶ Backup einzelner Datenfiles der CDB:

```
▪ RMAN> BACKUP DATAFILE 4,10;
```


Multitenant

Namensgebung Fast-Recovery-Area

- ▶ "GUID" ist statt des PDB-Namen im Pfad-Namen der Fast-Recovery-Area enthalten.
 - `handle=C:\ORACLE\FAST_RECOVERY_AREA\CDB\2176D7988B2A41FDB4DFFB91BC1603A4\BACKUPSET\2014_11_14\01_MF_NNND F_TAG20141114T103211_B6CM317B_.BKP tag=TAG2014`

- ▶ Abfrage zur Ermittlung der PDB:

- `SQL> SELECT CON_ID, DBID, CON_UID, GUID, NAME FROM v$pdb;`

- | CON_ID | DBID | CON_UID | GUID | NAME |
|--------|------------|------------|----------------------------------|-----------|
| 2 | 339868616 | 339868616 | 7855C3AE48A847D5963E9D4E5A02A19C | PDB\$SEED |
| 3 | 3182451474 | 3182451474 | 0A100C9A070545B6A45B0862935AB1FC | PDB1 |
| 4 | 3368374370 | 3368374370 | 27758307B746450483BB5B2911BF5169 | PDB2 |

Multitenant

"RESTORE | RECOVERY ..." Beispiele für Wiederherstellung mit PDB

▶ Wiederherstellung der gesamten PDB

```
▪ % rman target sys/<password>@cdb
▪ RMAN> RESTORE PLUGGABLE DATABASE pdb1;
RMAN> RECOVER PLUGGABLE DATABASE pdb1;
RMAN> ALTER PLUGGABLE DATABASE pdb1 open;
```

▶ Wiederherstellung eines Non-System-Tablespaces der PDB

```
▪ % rman target sys/<password>@pdb1
▪ RMAN> ALTER TABLESPACE users OFFLINE;
RMAN> RESTORE TABLESPACE users;
RMAN> RECOVER TABLESPACE users;
RMAN> ALTER TABLESPACE USERS ONLINE;
```

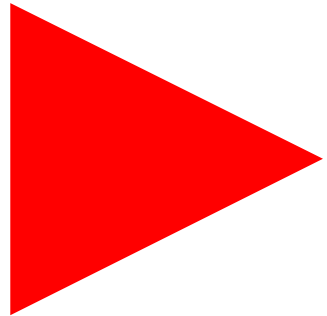
Multitenant

Database-Point-in-Time Recovery (DBPITR) mit PDBs

- ▶ Database-Point-in-Time Recovery (DBPITR) für PDBs
 - Unvollständige Wiederherstellung einer PDB
 - Optionen: "UNTIL TIME | SCN | SEQUENCE"
 - "In-Place"-Wiederherstellung der Datenfiles der PDB an der ursprünglichen Lokation.
 - Auxiliary-Datenbank ("Automatic Instance") wird für die weitere Wiederherstellung verwendet
 - Aufgrund der gemeinsamen Ressourcen-Nutzung: Tablespaces UNDO, SYSTEM, SYSAUX und SYSEXT werden im Root der Auxiliary-Datenbank wiederhergestellt.

- ▶ Beispiel:

```
% rman target sys/<password>@cdb
ALTER PLUGGABLE DATABASE pdb1 CLOSE;
run
{
  SET UNTIL SCN 969696;
  RESTORE PLUGGABLE DATABASE pdb1;
  RECOVER PLUGGABLE DATABASE pdb1;
}
ALTER PLUGGABLE DATABASE pdb1 OPEN RESETLOGS;
```

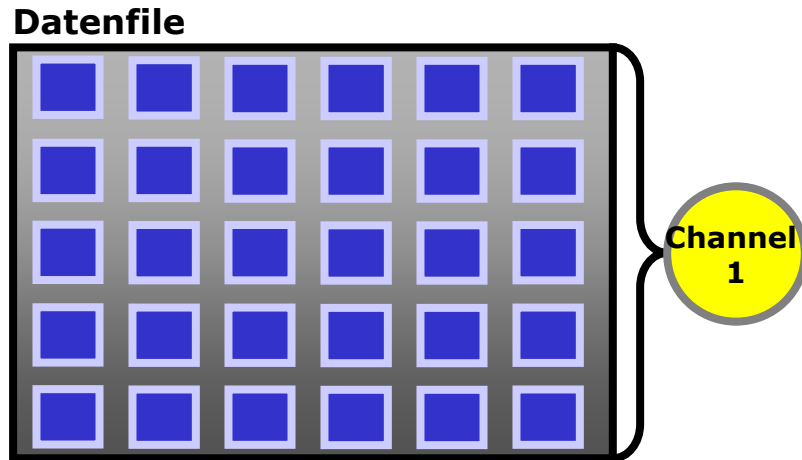


Multisection-Backups

ORACLE®
DATABASE **12^c**

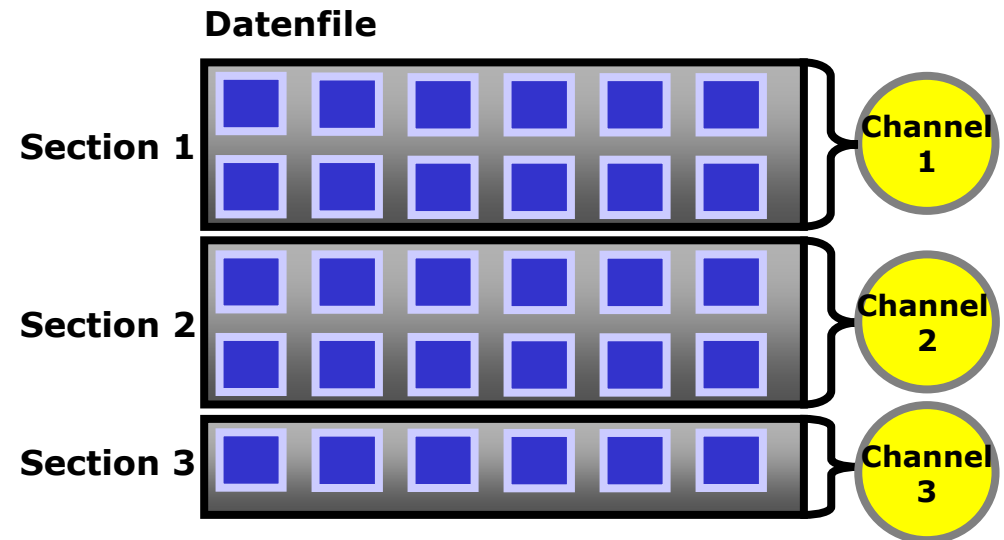
Multisections-Backups

10g: Ohne Multisections



- ▶ Limitierte Skalierbarkeit:
Datenfile ist kleinste
Verarbeitungseinheit pro Channel
- ▶ RMAN> BACKUP DATABASE;

11g: Mit Multisections



- ▶ Sections: Gleichgroße Abschnitte aus
zusammenhängenden Blöcken
- ▶ RMAN> BACKUP **SECTION SIZE 2G**
DATABASE;

Multisections mit Image-Kopien und inkrementellen Backups

RMAN 12c

▶ Neu in 12c: Inkrementelle Backups mit Multisections

```
▪ RMAN> BACKUP INCREMENTAL LEVEL 1 SECTION SIZE 2G DATAFILE 12;
```

- "Unused-Block-Compression" und "Block-Change-Tracking" werden genutzt.
- Für Level 0 Multisection-Incremental-Backups: COMPATIBLE = 11.0+
- Für Level 1 Multisection-Incremental-Backups: COMPATIBLE = 12.0.0+
- Views: V\$BACKUP_SET | RC_BACKUP_SET, Spalte MULTI_SECTION

▶ Neu in 12c: Image-Kopien mit Multisections

```
▪ RMAN> BACKUP AS COPY SECTION SIZE 2G DATAFILE 12;
```

- Schnellere Erstellung von Image-Kopien, insbesondere für Transportable Tablespaces, Cloning und im Exadata-Umfeld
- Voraussetzung: COMPATIBLE = 12.0+



Active Database Duplication Improvements

ORACLE[®]
DATABASE **12^c**

Active Database Duplication

Überblick 11g: Push-Methode

- ▶ Funktionalität in **11g: Push-Methode**
 - Netzwerkbasierter Duplizierungsmechanismus, ohne Zwischenspeicherung des Backups.
 - Führt eine **Image-Kopie** der Datenbank via Netzwerk aus.
 - **Work-Prozesse/Target-Channels** werden von der **Quell-Datenbank** (Target) ausgeführt.
- ▶ Beispiel:

```
RMAN> DUPLICATE TARGET DATABASE TO test1db
      FROM ACTIVE DATABASE
      SPFILE
      PARAMETER_VALUE_CONVERT '/prod1db', '/test1db'
      SET DB_FILE_NAME_CONVERT '/prod1db', '/test1db'
      SET LOG_FILE_NAME_CONVERT '/prod1db', '/test1db'
      SET MEMORY_TARGET='400M';
```

- ▶ Restriktionen bzw. Nachteile in 11g
 - Kein Point-In-Time-Recovery (kein "RECOVER UNTIL TIME | SCN | SEQUENCE" möglich)
 - Overhead: Belastung von Quell-Datenbank & Netzwerk
 - Nur Image-Kopien möglich

Active Database Duplication

Neu in 12c: Pull-Methode

▶ 12c: Pull-Methode

- Einsatz von Backup-Sets
- Nutzung von Compressed-, Multisection-Backups und Verschlüsselung
- **Work-Prozesse/Auxiliary-Channels** werden auf der Ziel-Datenbank (Auxiliary) ausgeführt.
- Neues Schlüsselwort "USING [COMPRESSED] BACKUPSET"
- Neue Default-Methode in 12c / empfohlene Methode

▶ Beispiel:

```
RMAN> DUPLICATE TARGET DATABASE TO test1db
FROM ACTIVE DATABASE
USING COMPRESSED BACKUPSET
SECTION SIZE 2G
SPFILE
    PARAMETER_VALUE_CONVERT '/prod1db', '/test1db'
SET DB_FILE_NAME_CONVERT '/prod1db', '/test1db'
SET LOG_FILE_NAME_CONVERT '/prod1db', '/test1db'
SET MEMORY_TARGET='400M';
```

Active Database Duplication

Neu in 12c: Pull-Methode

▶ Nice to know:

- Für die Parallelisierung müssen mehrere **Auxiliary-Channels** angegeben werden.
- Anmeldung an Target erfolgt weiterhin an der Quell-Datenbank.
- Es werden bei der Ausführung **immer neue Backup-Sets** erzeugt; bereits bestehende Backup-Sets können nicht genutzt werden.
- In 12c wird weiterhin auch die 11g-Push-Methode unterstützt.

▶ Vorteile:

- Weniger Belastung der Quell-Datenbank (Target).
- In Verbindung mit Backup-Sets können Compressed-, Multisection-Backups und auch Verschlüsselung eingesetzt werden.
- Unused-Block-Compression wird verwendet.
- Schnellere Ausführung / besserer Durchsatz bei der Netzwerkübertragung.

Active Database Duplication

Vergleich Push- und Pull-Methode

11g: Push-Methode

- ▶ Work-Prozesse/Target-Channels auf der Quell-Datenbank (Target)
- ▶ Nur mit Image-Kopien möglich

12c: Pull-Methode (Default)

- ▶ Work-Prozesse/Auxiliary-Channels auf der Ziel-Datenbank (Auxiliary)
- ▶ Mit Backup-Sets: "USING BACKUPSET"
 - Compressed-Backups: "USING COMPRESSED BACKUPSET"
 - Multisection-Backups: "SECTION SIZE"
 - Verschlüsselung
 - Unused-Block-Compression
 -
- ▶ Vorteil:
 - Schnellere Übertragung / höherer Durchsatz über Netzwerk
 - Geringere Belastung der Quell-Datenbank
 - Gleichmäßigere Verteilung des Loads auf der Auxiliary-DB

DUPLICATE ohne RESETLOGS beim Öffnen der duplizierten Datenbank

"NOOPEN"-Option

- ▶ 11g: Auxiliary-Datenbank wurde nach dem Duplicate abschließend automatisch mit der "RESETLOGS"-Option geöffnet.

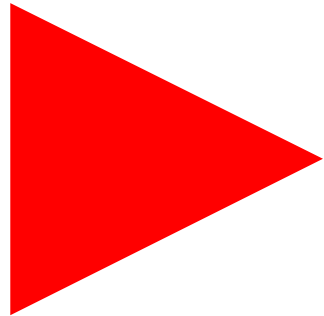
- ▶ Neu in 12c: **"NOOPEN"-Option**

- "DUPLICATE ... NOOPEN" belässt die Auxiliary-Datenbank im Status "MOUNT"

```
RMAN> DUPLICATE TARGET DATABASE TO testdb
FROM ACTIVE DATABASE NOOPEN;
#--> Hier Anpassungen durchführen <--#
ALTER DATABASE OPEN [UPGRADE];
```

- ▶ Anwendung: Für Anpassungen

- Eigenschaften der Datenbank ändern: Flashback-Database-Logs oder Block-Change-Tracking-Files konfigurieren.
- Bei Upgrade-Szenarien, wenn die duplizierte Datenbank in den "OPEN UPGRADE"-Mode gebracht werden soll.



Cross-Platform Backup & Restore Improvements

ORACLE®
DATABASE **12^c**

"Cross-Plattform"-Verfahren

Überblick: Bis 11g

- ▶ **Cross-Platform Transportable Tablespaces (seit 10g):** "CONVERT TABLESPACE ..."
 - Plattformübergreifender Transport von Tablespaces bzw. Datenfiles
 - **Endianess-Konvertierung** möglich
 - Export/Import von Metadaten notwendig
 - Tablespace muss während des Kopierens "READ ONLY" sein.

- ▶ **Cross-Platform Transportable Database (seit 10gR2):** "CONVERT DATABASE ..."
 - Plattformübergreifender Transport der gesamten Datenbank
 - **Keine Endianess-Konvertierung** möglich
 - Kein Export/Import von Metadaten notwendig
 - Datenbank muss während des Kopierens "READ ONLY" sein.

- ▶ Resriktionen bis 11g:
 - Für den den plattformübergreifenden Transport können nur **Image-Kopien** eingesetzt werden.
 - "**READ ONLY**"-Modus hat **Implikationen auf Verfügbarkeit der Applikation.**

Cross-Platform Backup & Restore mit Backup-Sets

Neu in 12c

- ▶ **Cross-Platform Backup & Restore mit Backup-Sets (seit 12c): "BACKUP ... | RESTORE..."**
 - Transport-Ebenen:
 - **Gesamte Datenbank** (inkl. SYTEM- und SYSAUX-Tablespace)
 - Einzelne **Tablespaces und Datenfiles**
 - **Backup-Sets** werden für den plattformübergreifenden Transport eingesetzt
 - Tablespaces können im "**READ WRITE**"-Modus inkonsistent gesichert werden.
 - Durch inkrementelle Backups kann der "READ ONLY"-Modus zeitlich erheblich minimiert werden.
 - Endianess-Konvertierung:
 - Bei gesamter **Datenbank**: **Keine Endianess-Konvertierung** möglich
 - Bei einzelnen **Tablespaces** bzw. Datenfiles: **Endianess-Konvertierung** möglich
 - Plattform-Konvertierung erfolgt
 - Auf der **Quell-Datenbank** beim Backup
 - Auf der **Ziel-Datenbank** beim Restore
- ▶ Vorteile: Prädestiniert für heterogene Plattform-Migrationen.
 - Im Gegensatz zu "Cross-Platform Transportable Tablespace" und "Cross-Platform Transportable Database" ist die Migration wesentlich **einfacher** und eleganter durchführbar.
 - **Schnellere Transportzeiten**: Durch Backup-Sets können u.a. Compressed- und Multisection-Backups eingesetzt werden.

Cross-Platform Backup & Restore mit Backup-Sets

"BACKUP FOR TRANSPORT | TO PLATFORM ..." Beispiele

- ▶ Mit "FOR TRANSPORT" wird ein generisches Cross-Platform-Backup ohne Konvertierung erzeugt:

```
▪ RMAN> BACKUP FOR TRANSPORT  
  FORMAT '/u03/crossover/db_migration.rman' DATABASE;
```

- ▶ Mit "TO PLATFORM='...'" wird das Backup auf der Quell-Datenbank konvertiert:

```
▪ RMAN> BACKUP TO PLATFORM='Linux x86 64-bit'  
  FORMAT 'c:\crossover\db_migration.rman' DATABASE;
```

- ▶ View für unterstützte Plattformen: V\$TRANSPORTABLE_PLATFORM

Cross-Platform Backup & Restore mit Backup-Sets

"RESTORE FROM PLATFORM | ALL FOREIGN DATAFILES" Beispiele

- ▶ Mit "FROM PLATFORM '...' FOREIGN DATABASE TO NEW" wird während des Restores eine Konvertierung durchgeführt:

```
▪ RMAN> RESTORE FROM PLATFORM 'microsoft windows x86 64-bit'  
FOREIGN DATABASE TO NEW  
FROM BACKUPSET '/u03/crossover/db_migration.rman';
```

- Über die Klausel "FOREIGN DATABASE TO NEW" werden alle Objekte – also die gesamte Datenbank – spezifiziert.

- ▶ Mit "ALL FOREIGN DATAFILES" wird ein bereits während des Backups konvertiertes Backup auf der Ziel-Plattform wiederhergestellt:

```
▪ RMAN> RESTORE ALL FOREIGN DATAFILES  
FORMAT '/u03/oradata/datafile_%U'  
FROM BACKUPSET '/u03/crossover/db_migration.rman';
```



Restoring & Recovering Files Over a Network

ORACLE[®]
DATABASE **12^c**

Restoring & Recovering Files Over a Network

"RESTORE | RECOVER ... FROM SERVICE"

- ▶ Mit 11g im DataGuard-Verbund: **Austausch von Controlfiles & Backups** sowie **Topologie-Unterstützung**
 - RMAN-Backups sowie Controlfiles konnten beidseitig zwischen der Primär- und der Physical Standby-Datenbank zur Wiederherstellung ausgetauscht werden.
 - Topologie-Unterstützung: Zugänglichkeit von Backups (Tape/Disk) wird bei der Wiederherstellung beachtet.
- ▶ Mit 12c:
 - Physical-Standby-Datenbanken können jetzt direkt über Netzwerk ("**FROM SERVICE**") für ein Restore & Recovery der Primär-Datenbank (und umgekehrt) herangezogen werden.
 - Für die Wiederherstellung werden automatisch **Backup-Sets** angefertigt, die über Oracle-Net zum Ziel-Host transferiert und dann wiederhergestellt werden.
 - Wiederherstellung von Datenbank, Tablespaces, Datenfiles und Controlfiles
 - Unterstützung von Multisection-Backups, Compression und Verschlüsselung.

Restoring & Recovering Files Over a Network

"RESTORE | RECOVER ... FROM SERVICE"

- ▶ **Wiederherstellung von Datenfiles** über das Netzwerk über die Primär- oder Standby-Datenbank

```
▪ RMAN> RESTORE DATAFILE '/u03/oradata/users01.dbf'  
FROM SERVICE <standby_service_tns>  
SECTION SIZE 2G;
```

- ▶ **Synchronisation / Roll-Forward** der Standby-Datenbank mit der Primär-Datenbank über ein inkrementelles Backup

```
▪ RMAN> RECOVER DATABASE FROM SERVICE <primary_service_tns>  
NOREDO SECTION SIZE 2G;
```

- Eine Synchronisation über inkrementelle Backups ist beispielsweise dann erforderlich, wenn NOLOGGING-Operationen auf der Primär-Datenbank stattgefunden haben, die nicht über den Redo-Log-Stream an die Standby-Datenbank übertragen wurden.



Recovering Tables & Table Partitions from RMAN Backups

ORACLE[®]
DATABASE **12^c**

Wiederherstellung von Tabellen und Tabellen-Partitionen aus RMAN-Backups

"RECOVER TABLE"

► Funktionalität

- Point-In-Time-Recovery von Tabellen bzw. Tabellen-Partitionen aus bestehenden RMAN-Backups.
- Logische Wiederherstellung über ein physikalisches Backup.
- Sinnvoll, wenn die Daten nicht mehr via Flashback wiederhergestellt werden können, die Tabelle gelöscht wurde oder logisch korruptiert ist.

► Die Wiederherstellung mit Point-In-Time-Recovery auf Tabellenebene kann altbewährt mit "UNTIL TIME | SCN | SEQUENCE" erfolgen.

► Beispielsweise:

```
RMAN> RECOVER TABLE SCOTT.EMP, SCOTT.DEPT
UNTIL TIME 'SYSDATE-1'
AUXILIARY DESTINATION '/u03/oradata/aux'
DATAPUMP DESTINATION '/u03/pumps'
DUMP FILE 'exp_dump.exp'
NOTABLEIMPORT;
```

Wiederherstellung von Tabellen und Tabellen-Partitionen aus RMAN-Backups

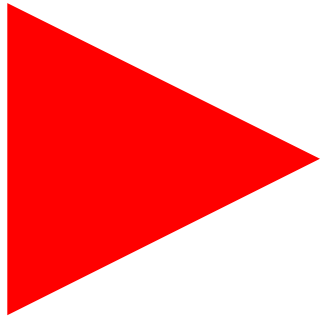
"RECOVER TABLE"

▶ Grundlegender Ablauf:

- RMAN erzeugt für das Point-In-Time-Recovery automatisiert eine Auxiliary-Datenbank
- Point-In-Time-Recovery des Tablespaces mit den gewünschten Tabellen.
- Erzeugen eines DataPump-Exportes mit den spezifizierten Objekten
- Optional: Import der Tabellen bzw. Partitionen mit Data-Pump in die Ziel-Datenbank
- Optional: Remappings für Tabellen und Tablespaces

▶ Methode: Ein logisches Backup?

- Die Methode kann als verbessertes TSPITR-Verfahren betrachtet werden und ist feingranularer direkt auf Objekte bezogen und daher schneller umsetzbar.
- Ein "BACKUP TABLE ..." -Kommando für eine logische Sicherung auf Objekt-Ebene existiert daher nicht.



More Features!

ORACLE[®]
DATABASE **12^c**

Interface Verbesserungen & "SYSBACKUP"-Privileg

▶ Interface Verbesserungen

- SQL- und PL/SQL-Kommandos können direkt im RMAN-Prompt ausgeführt werden: Ohne "SQL '<sql_statement>'"-Kommando
- "SELECT" und "DESCRIBE"-Kommando sind implementiert

▶ Neues "SYSBACKUP"-Privileg

- Speziell für Backup- und Recovery-Aufgaben
- Keine Berechtigung für den Zugriff auf Tabellen-Daten (Kein "SELECT ANY TABLE"-Privileg)

Storage Snapshot Optimization

"SNAPSHOT TIME"

▶ Funktionalität

- 3rd-Party-Storage-Technologien für Storage-Snapshots der Datenbank
- Kein BACKUP-Mode ("ALTER DATABASE BEGIN|END BACKUP") erforderlich
- Storage-Snapshot muss bestimmten Anforderungen entsprechen
- Neues Schlüsselwort "SNAPSHOT TIME" für das RECOVER-Kommando

- Beispiel: Wiederherstellung der Datenbank mit Snapshot:

```
– RMAN> RECOVER DATABASE UNTIL TIME '11/18/2014 16:00:00'  
        SNAPSHOT TIME '11/18/2014 15:00:00';
```

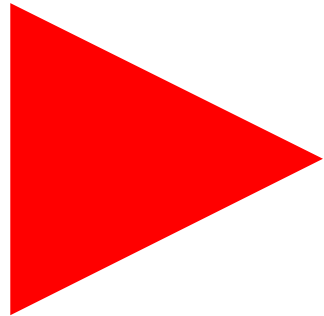
▶ Vorteile

- Einfache Anwendung, ohne weitere manuelle Schritte
- Kein I/O-Overhead durch Full-Block-Logging

Unified Auditing & RMAN Virtual Private Catalog

- ▶ Unified Auditing
 - Unified Auditing konsolidiert Audit-Trails
 - View UNIFIED_AUDIT_TRAIL enthält Felder mit Prefix "RMAN_".

- ▶ Oracle Virtual Private Database (VPD) für RMAN Virtual-Private-Catalog [ab 12.1.0.2]
 - Virtual-Recovery-Catalog nutzt Virtual Private Database (VPD)-Feature
 - Enterprise Edition für RMAN-Catalog zwingend erforderlich



Zusammenfassung & Resümee

ORACLE®
DATABASE **12^c**

Zusammenfassung & Resümee

***"Das Bessere ist der Feind des Guten."* (Voltaire, 1694-1778)**

- ▶ Integration & Erweiterung von RMAN in die neue Multitenant-Architektur.
- ▶ Optimierung und Perfektionierung bekannter Features.
- ▶ Logisches Restore wird für Tabellen und Tabellen-Partitionen eröffnet. Hier ist zu erhoffen, dass es zukünftig Möglichkeiten für die logische Sicherung und Wiederherstellung von Objekten mit RMAN geben könnte.
- ▶ Highlights von Oracle Database 12c sind:
 - "Cross-Platform Backup & Restore"
 - "Restore & Recovery über das Netzwerk mit Services".
- ▶ Integration von SQL*Plus-Funktionalitäten in das RMAN-Interface vereinfachen das "Daily Life" des DBAs.

Quellen

▶ Oracle Documentation

- Oracle Database Backup and Recovery User's Guide 12c Release 1 (12.1), E50658-05, September 2014
- Oracle Database Backup and Recovery Reference 12c Release 1 (12.1), E50791-04, September 2014
- Oracle Database New Features Guide 12c Release 1 (12.1), E49322-08, July 2014

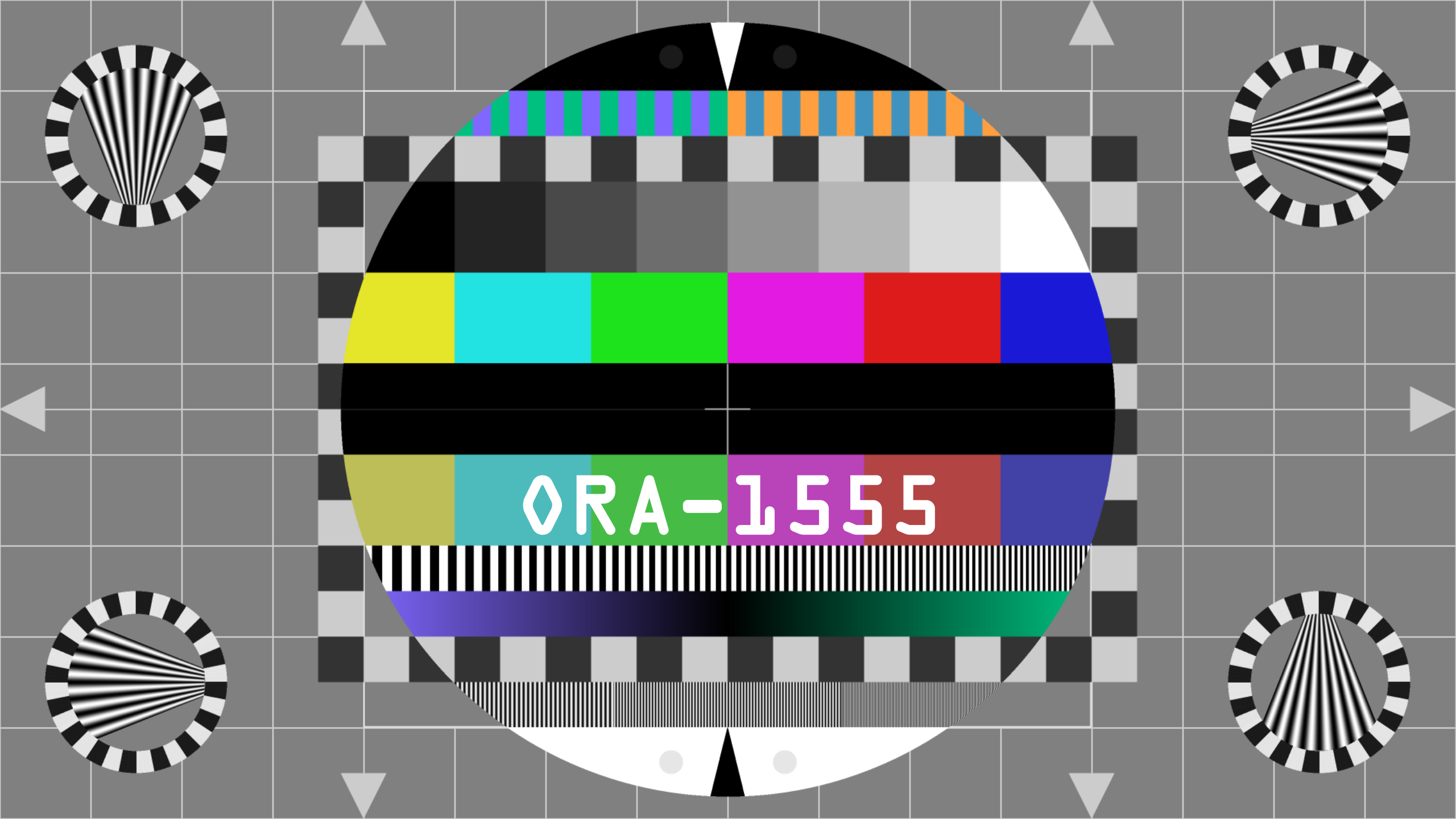
▶ MOS-Notes

- RMAN Enhancements in Oracle 12c (Doc ID 1534487.1)
- RMAN Pluggable Database Point in Time Recovery (Note 1521075.1)
- RMAN Pluggable Database Backup and Recovery (Note 1521005.1)
- RMAN RECOVER TABLE Feature New to Oracle Database 12c (Note 1521524.1)

RMAN Reloaded

**Recovery Manager
News & Features** **12^c**

Thank You!



ORA-1555