



# Backup und Recovery

Johannes Ahrends  
CarajanDB GmbH



- **Experten mit über 20 Jahren Oracle Erfahrung**
- **Firmensitz in Erftstadt bei Köln**
- **Spezialisten für**
  - Oracle Datenbank Administration
  - Hochverfügbarkeit (RAC, Data Guard, Failsafe, etc)
  - Einsatz der Oracle Standard Edition
  - Oracle Migrationen (HW, Unicode, Konsolidierung, Standard Edition)
  - Replikation (Goldengate, SharePlex, Dbvisit)
  - Performance Tuning
- **Schulung und Workshops (Oracle, Toad)**



- **Oracle Spezialist seit 1992**
  - 1992: Presales bei Oracle in Düsseldorf
  - 1999: Projektleiter bei Herrmann & Lenz Services GmbH
  - 2005: Technischer Direktor ADM Presales bei Quest Software GmbH
  - 2001: Geschäftsführer CarajanDB GmbH
- **2011 → Ernennung zum Oracle ACE**
- **Autor der Bücher:**
  - Oracle9i für den DBA, Oracle10g für den DBA, Oracle 11g Release 2 für den DBA
- **Leiter der DOAG SIG Database**
- **Hobbies:**
  - Drachen steigen lassen (Kiting) draußen wie drinnen (Indoorkiting)
  - Motorradfahren (nur draußen)



- E-Mail: [johannes.ahrends@carajandb.com](mailto:johannes.ahrends@carajandb.com)
- Homepage: [www.carajandb.com](http://www.carajandb.com)
- Adresse:
  - CarajanDB GmbH  
Siemensstraße 25  
50374 Erftstadt
- Telefon:
  - +49 (22 35) 1 70 91 84
  - +49 (1 70) 4 05 69 36
- Twitter: [streetkiter](#)
- Facebook: [johannes.ahrends](#)
- Blogs:
  - [www.carajandb.com/blogs](http://www.carajandb.com/blogs)
  - [streetkiter.wordpress.com](http://streetkiter.wordpress.com)
  - [www.toadworld.com](http://www.toadworld.com)



# Backup und Recovery

Zitat (wikipedia):

Datensicherung (englisch backup ['bækʌp]) bezeichnet das Kopieren von Daten in der Absicht, diese im Fall eines Datenverlustes zurückkopieren zu können.

Die auf dem Speichermedium gesicherten Daten werden als Sicherungskopie, engl. Backup, bezeichnet. Die Wiederherstellung der Originaldaten aus einer Sicherungskopie bezeichnet man als Datenwiederherstellung, Datenrücksicherung oder (englisch) Restore.

# Warum Datenverlust?

- Plattencrash
- Servercrash (?)
- Storageausfall
- Stromausfall (?)
- Rechenzentrumsausfall

# Wie schützt man sich davor?

- Plattencrash → Plattenspiegelung (Raid 1, ...)
- Servercrash → zweiter Server (z.B. Virtualisierung, Cluster)
- Storageausfall → Storgespiegelung (EMC SRDF, etc.)
- Stromausfall → USV
- Rechenzentrumsausfall → zweites Rechenzentrum (Desaster Recovery)

# Welche Fehler gibt es noch?

- **Anwendungs- bzw. Anwenderfehler**
  - Fehlerhafte Software (z.B. nach Releasewechsel)
  - Fehlerhafte Dateneingabe / Änderung
  - Vorsätzliche Zerstörung von Daten
- **Bugs**
  - Datenbank schreibt falsche Daten
  - Controller defekt

# Wie schützt man sich davor?

- **Anwendungs- bzw. Anwenderfehler**

- Fehlerhafte Software (z.B. nach Releasewechsel) → Testen
- Fehlerhafte Dateneingabe / Änderung → Anwendungsprogrammierung
- Vorsätzliche Zerstörung von Daten → Sicherheitsbewusstsein

- **Bugs**

- Datenbank schreibt falsche Daten → Replikation der Daten
- Controller defekt → logische Spiegelung der Daten

Grundsätzlich gilt:

**Es gibt für jede Lösung einen Fehler!**

# Was wird gesichert?

- **Tabellendefinition und –inhalt**
- **Datenbank-Dateien**
- **Festplatte bzw. Partition**
- **Transaktionsprotokolle (Redolog)**

# Wann wird gesichert

- Einmal pro Tag
- Einmal pro Stunde
- Einmal pro Minute
- Vor bestimmten Änderungen
- Nach bestimmten Änderungen
- Aufgrund von bestimmten Kriterien  
(z.B. wie viele Daten sind in der Zwischenzeit angefallen)

# Wie wird gesichert?

- **Offline Backup**
  - Die Anwendung wird beendet
  - Alle relevanten Daten werden gesichert
  - Auszeit der Anwendung von der Größe der Datenbank abhängig
- **Online Backup**
  - Die Datenbank wird gesichert, während Änderungen durchgeführt werden
  - Diese Sicherung ist zunächst einmal inkonsistent

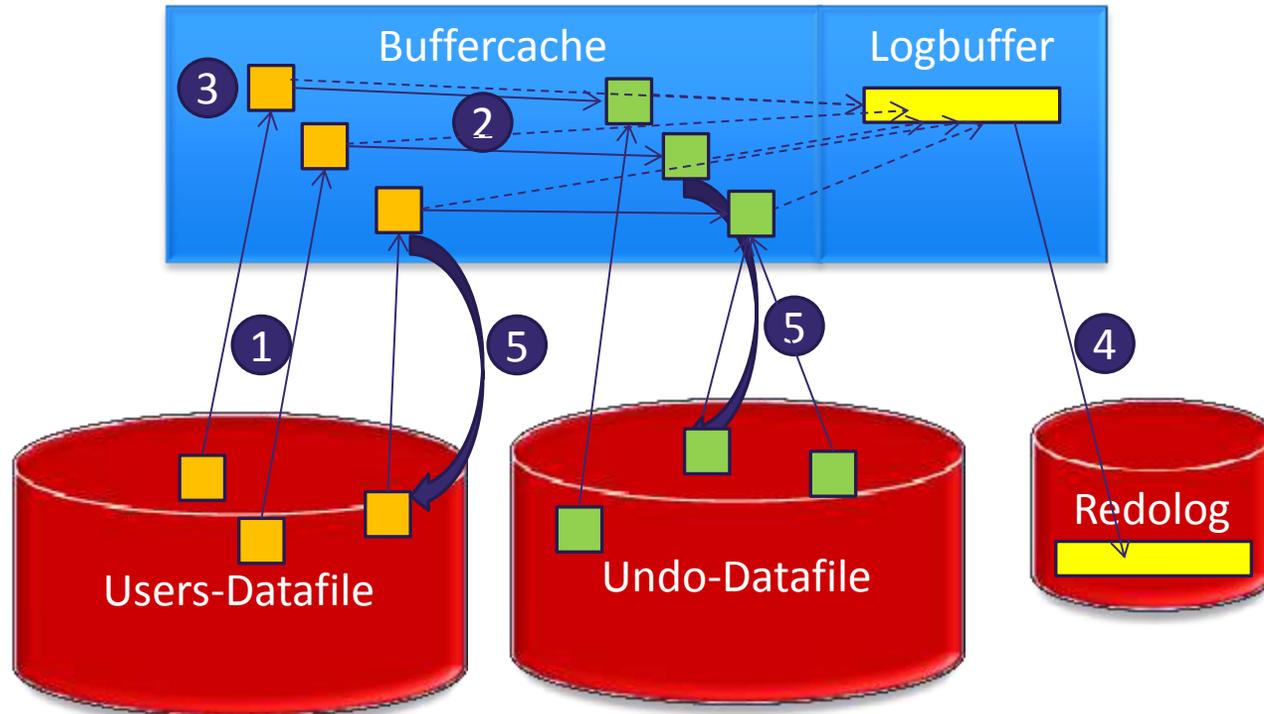
- **RPO = Recovery Point Objective**
  - Bis zu welchem Punkt kann ich Daten wieder herstellen?
  - D.h. wie viel Datenverlust kann ich tolerieren?
- **RTO = Recovery Time Objective**
  - Wie lange dauert die Wiederherstellung?
  - D.h. wie lange steht die Anwendung nicht zur Verfügung?

- 12:00 Start der Sicherung
- 12:05 Anwender 1 gibt folgenden Befehl ein:  

```
SQL> UPDATE adresse SET PLZ=40000; COMMIT;
```
- 13:00 Ende der Sicherung
  
- Frage: welche PLZ hat jede Adresse?
- Antwort: keine Ahnung!

# Was passiert bei einer Änderung

- UPDATE adresse SET  
PLZ=40000 ;
- COMMIT ;



## UPDATE:

1. Lesen der Daten als Oracle Blöcke aus den Datafiles
2. Kopieren der „alten“ Werte in die Undo-Segmente + Eintragen der Werte in den Logbuffer
3. Eintragen der „neuen“ Werte (PLZ=40000) in die Blöcke + Eintragen der Werte in den Logbuffer

## COMMIT;

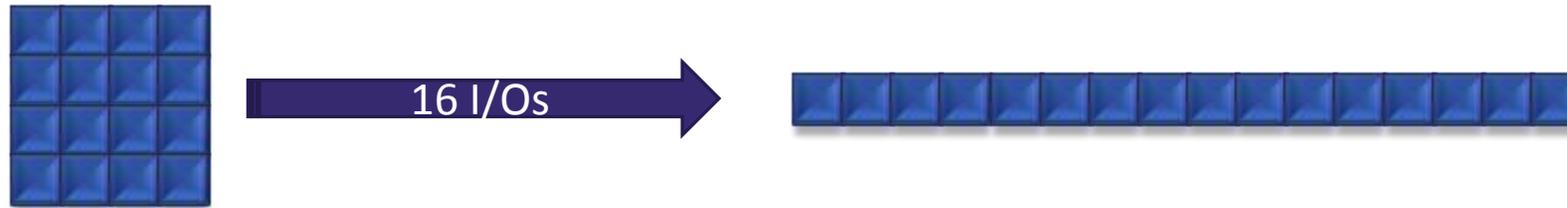
4. Schreiben des Logbuffers in die Redologdatei
- Später:
5. Schreiben der Undo und User Blöcke in die Datafiles

# Wodurch wird die Sicherung konsistent?

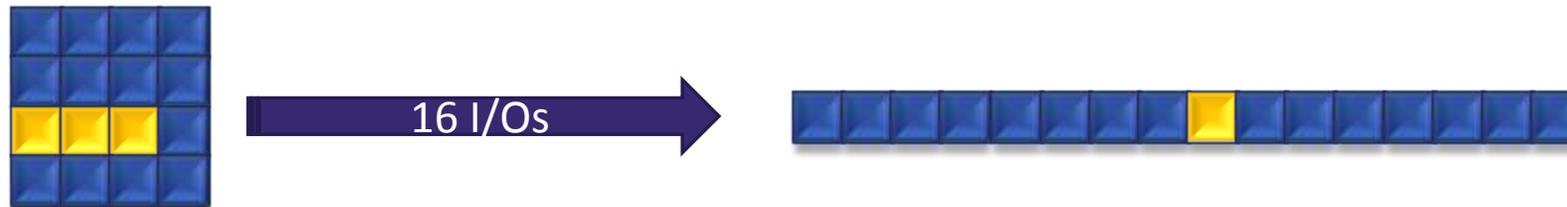
- **Sichern der Datendateien**
  - Zunächst einmal inkonsistent (wurden die „neuen“ oder die „alten“ Blöcke gesichert?)
- **Sichern der Undodateien**
  - Immer noch inkonsistent (aber jetzt gibt es sowohl „alte“ als auch „neue“ Werte!)
- **Sichern der Redolog-Dateien**
  - Konsistent, denn hieraus kann der Buffercache rekonstruiert werden (Recovery)

- „Wir sichern einfach alle Filesysteme / Partitionen mit unserem Backup Tool (z.B. Acronis oder einfach per Copy).“
- Warum funktioniert das nicht?

- Oracle Block → zwischen 2k und 32k (Standardmäßig 8k)
- Betriebssystemblock → zwischen 512 Byte und 1 MB
- Beispiel: Oracle Block = 8k, OS Block = 512 Byte



- `SQL> UPDATE orte set PLZ=40000 WHERE persid=1;`

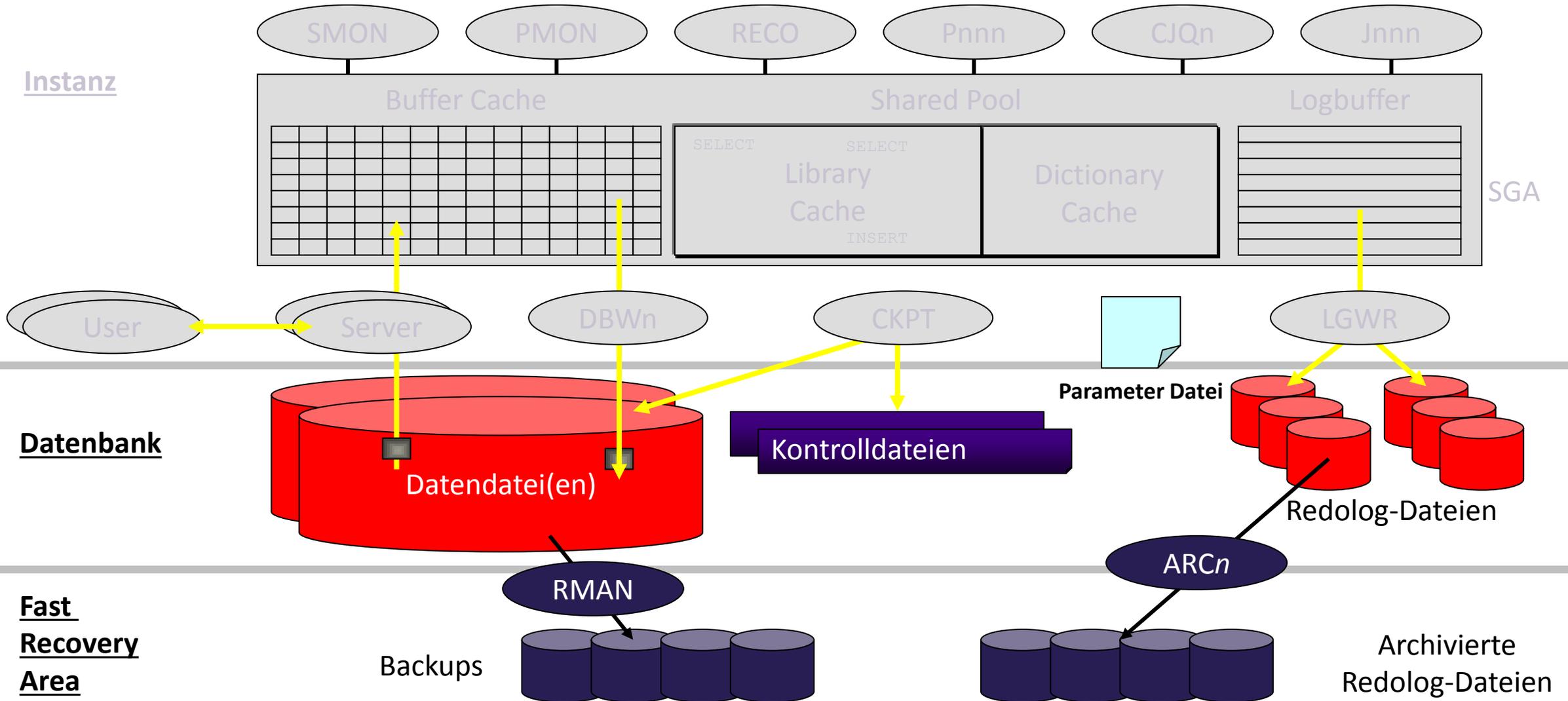


- Der Inhalt des Oracle Blockes ändert sich während der Sicherung → Der Block ist korrupt!

- **NIE mit Betriebssystemmitteln eine Online Oracle Datenbank sichern!**
  - Ausnahme: Die Datenbank bzw. die Datendatei wurde vorher in den „BEGIN BACKUP“ Modus gesetzt
- **Ausschließlich Oracle Backup Verfahren = RMAN verwenden!**
  - RMAN prüft, ob sich ein Oracle Block während der Sicherung geändert hat und schreibt dann einfach den Block noch einmal

- **DB-Vorbereitung**
- **Archiving**
- **Backup Strategie**
- **Recovery Manager (RMAN)**
- **Fehleranalyse**
- **Recovery-Strategien**
- **Weitere Technologien**

# Backup Komponenten



- **Erster Level der Verfügbarkeit einer Datenbank**
  - sollte immer aufgesetzt sein
- **Hardware oder Betriebssystemspiegelung**
  - Raid 1, 3 ,5 ...
- **Spiegelung der Datenbank oder Teile davon**
- **Dadurch wird ein Backup nicht überflüssig!**
- **Ab Raid 3 Performanceeinbußen beim Schreiben**
- **Kontroll- und Redolog-Dateien sollten trotzdem über Oracle Mechanismen gespiegelt werden**

# Redo Log Dateien spiegeln

- zweites (oder mehr) zusätzliches Set von Redo Log Dateien
- Werden über den Logwriter (LGWR) synchron beschrieben
- Ausfall einer Gruppe führt nicht zum Fehler

```
SQL> ALTER DATABASE  
      ADD LOGFILE THREAD 1 GROUP 3 '/oradata1/SUNDB/redo03.log',  
      '/oradata1/SUNDB/redo13.log' SIZE 20M;
```

```
SQL> ALTER DATABASE  
      ADD LOGFILE MEMBER  
      '/oradata1/SUNDB/redo13.log' TO GROUP 3;
```

```
SQL> ALTER DATABASE  
      ADD LOGFILE MEMBER  
      '/oradata1/SUNDB/redo13.log' TO '/oradata1/SUNDB/redo03.log';
```

# Kontrolldatei spiegeln

- **Kontroll-Datei spielt extrem wichtige Rolle für ein vollständiges Recovery!**
- **Backup der Kontroll-Datei bei jeder Struktur-Änderung**

# Kontrolldatei nachträglich spiegeln

- **SPFILE ändern:**

```
SQL> ALTER SYSTEM SET control_files =  
      '$HOME/ORADATA/u01/ctrl01.ctl',  
      '$HOME/ORADATA/u02/ctrl02.ctl'  
SCOPE=SPFILE;
```

- **Datenbank herunter fahren**

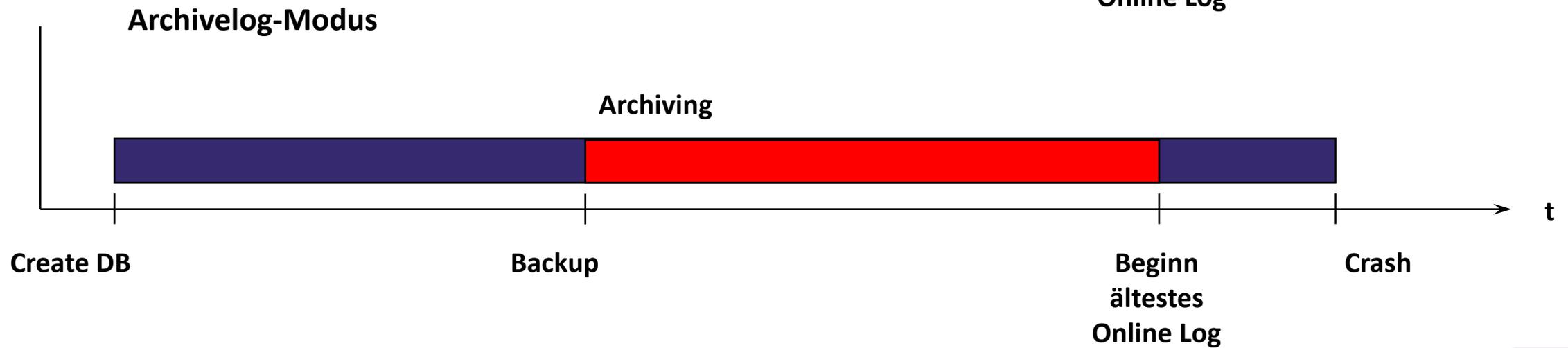
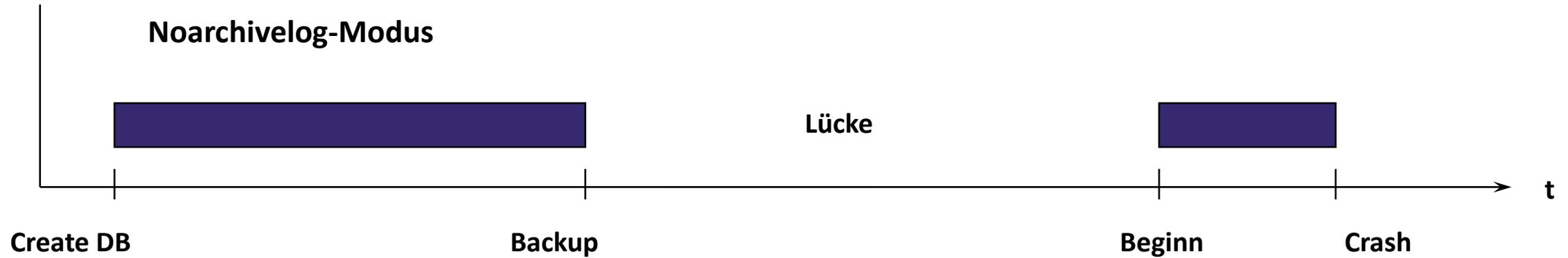
```
SQL> SHUTDOWN IMMEDIATE;
```

- **Zusätzliche Kontrolldateien kopieren**

```
$ cp $HOME/ORADATA/u01/ctrl01.ctl  
   $HOME/ORADATA/u02/ctrl02.ctl;
```

- **Datenbank wieder starten**

# Die Recovery-Lücke



# Redolog-Informationen behalten!

- Um eine Recoverylücke zu vermeiden, müssen die Redolog-Informationen permanent gesichert werden
- Oracle ARCHIVELOG Modus
- Wenn eine Redolog-Datei „voll“ ist, wird eine Kopie erstellt (archiviert)
- Bei einem Online Backup sollten die archivierten Redolog-Dateien, die bis zu diesem Zeitpunkt angefallen sind, mitgesichert werden

- **Wohin soll archiviert werden?**
  - Standardmäßig in eine definierte „Fast-Recovery-Area“
  - Weitere Ziele (bis zu 31) können definiert werden (LOG\_ARCHIVE\_DEST\_*n*)
- **Wann soll archiviert werden?**
  - Standardmäßig, wenn eine Redolog-Datei voll ist
  - Alternativ über den Parameter ARCHIVE\_LAG\_TARGET (z.B. alle 15 Minuten)

- **Status abfragen**

```
SQL> ARCHIVE LOG LIST
Database log mode          No Archive Mode
Automatic archival        Disabled
Archive destination       USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence 103
Current log sequence      105
```

- **Was ist „USE\_DB\_RECOVERY\_FILE\_DEST“?**

```
SQL> SHOW PARAMETER recovery_file

NAME                                TYPE                VALUE
-----
db_recovery_file_dest                string              D:\orabackup\fast_recovery_area
db_recovery_file_dest_size           big integer        10000M
```

# Archivelog Modus einschalten

```
SQL> SHUTDOWN IMMEDIATE

SQL> STARTUP MOUNT

SQL> ALTER DATABASE ARCHIVELOG;

SQL> ALTER DATABASE OPEN;

SQL> ARCHIVE LOG LIST
Database log mode                Archive Mode
Automatic archival              Enabled
Archive destination             USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence      103
Next log sequence to archive    105
Current log sequence            105
```

- **Spezieller Datenbank-Bereich für alles was mit Backup zu tun hat bzw. für Recovery benötigt wird:**
  - Archivierte Redolog-Dateien
  - RMAN-Backups der Datenbank
  - Flashback Logs
  - Implizit: `log_archive_dest_10=use_db_recovery_file_dest` (Oracle 10g)  
bzw. `log_archive_dest_1=use_db_recovery_file_dest` (ab Oracle 11g)
- **Angegeben wird dieser Bereich über zwei Serverparameter**
  - `db_recovery_file_dest` Verzeichnis (am besten eigenes Filesystem)
  - `db_recovery_file_dest_size` Maximale Nutzung durch diese Datenbank und damit automatische Überwachung

# Ändern / Anpassen Fast Recovery Area

- **Aktivierung und Prüfung durch**

```
SQL> ALTER SYSTEM SET db_recovery_file_dest = '/oraflash';
```

```
SQL> ALTER SYSTEM SET db_recovery_file_dest_size = 1G;
```

```
SQL> SELECT * FROM v$recovery_file_dest;
```

NAME	SPACE_LIMIT	SPACE_USED	SPACE_RECLAIMABLE	NUMBER_OF_FILES	CON_ID
D:\orabackup\fast_recovery_area	1048576000	0	0	0	0

# Platzprobleme mit Fast Recovery Area

- Wenn der angegebene Platz zu klein wird, bleibt die Datenbank u.U. stehen, falls dieser Bereich für die Archivelogs verwendet wird.
- Fehlermeldung im Alert-File

```
ORA-19815: WARNING: db_recovery_file_dest_size of 1048576000 bytes is 99.63% used, and has 3872768 remaining bytes available.
```

```
*****
```

```
You have the following choices to free up space from flash recovery area:
```

1. Consider changing your RMAN retention policy.  
If you are using dataguard, then consider changing your RMAN archivelog deletion policy.
2. Backup files to tertiary device such as tape using the RMAN command BACKUP RECOVERY AREA.
3. Add disk space and increase the db\_recovery\_file\_dest\_size parameter to reflect the new space.
4. Delete unnecessary files using the RMAN DELETE command.  
If an OS command was used to delete files, then use RMAN CROSSCHECK and DELETE EXPIRED commands.

```
*****
```

```
Deleted Oracle managed file  
/oraflash/JAUX11G/backupset/2004_08_06/o1_mf_nnndf_TAG20040806T175900_0k7bp7z3_.bkp
```

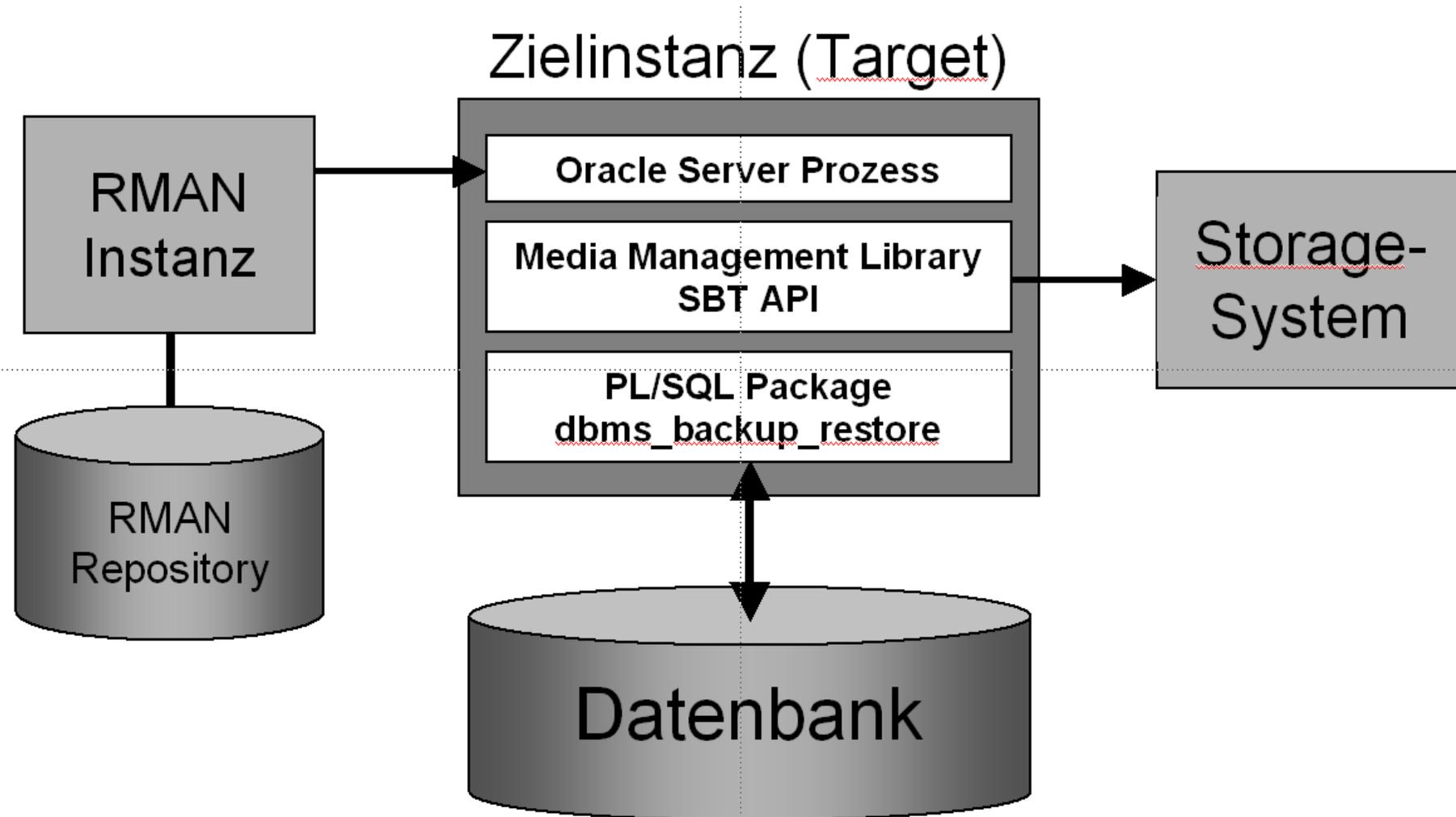
- **Online Database Backup**
  - Nur möglich, wenn die Datenbank im Archivelog-Modus gefahren wird
  - Vorteile:
    - Datenbank ist durchgängig verfügbar
- **Offline Database Backup**
  - im Archivelog- und im Noarchivelog-Modus möglich
  - Vorteile:
    - einfaches Konzept
  - Nachteile:
    - Datenbank ist während der Zeit des Backups nicht verfügbar
- **Für produktive Datenbanken immer Online Backup nutzen!**

- **SHUTDOWN** der Instanz
- **Sicherung der gesamten DB (Daten- und Kontroll-Dateien)**
  - Redolog-Dateien brauchen nicht gesichert zu werden
- **Sicherung hat konsistenten Zustand, falls die Instanz *NORMAL*, *IMMEDIATE* oder *TRANSACTIONAL* heruntergefahren ist**
  - ansonsten erfolgt beim nächsten *STARTUP* ein Instance Recovery (auch hierfür werden keine Redologs benötigt!)
- **Achtung: Es besteht immer die Gefahr, Dateien zu vergessen, daher:**
  - Datenbank nicht komplett offline sichern, sondern im Mount-Status
  - Zu sichernde Dateien auslesen via Views v\$datafile bzw. v\$controlfile

- Nur im *ARCHIVELOG*-Modus möglich und sinnvoll
- Instanz bleibt im *OPEN*-Status, d.h. Aktivität ununterbrochen
- **Sicherung wahlweise**
  - Der kompletten Datenbank
  - Einzelner Tablespaces
  - Einzelner Datendateien
- **Sollte immer den Oracle RMAN (Recovery Manager) nutzen**
- „Old-Style“ Online Backup sollte nicht mehr verwendet werden
  - ALTER TABLESPACE BEGIN | END BACKUP...

- **Transparentes, servergestütztes Backup und Recovery**
- **Protokollieren der Aktionen im Repository**
  - Kontrolldatei der Datenbank
  - Optional: zentrale „Katalog-Datenbank“
- **Automatisierung durch gespeicherte Skripts**
- **Aufbau von „Backup Sets“ oder einfachen „Image Copies“**
- **Differentielle Backup Sets: Kumulativ oder inkrementell**
- **Konsistenzprüfung (fractured block detection)**
- **Erkennen und Protokollieren von korrupten Blöcken**
- **Optional: direkte Anbindung an Tape-Libraries**
  - Tivoli Storage Manager, EMC Networker, Symantec NetBackup etc.

# Recovery Manager Architektur



# RMAN Backup Syntax

```
RMAN> BACKUP DATABASE;  
  
RMAN> BACKUP ARCHIVELOG ALL;  
  
RMAN> BACKUP DATABASE PLUS ARCHIVELOG;  
  
RMAN> BACKUP TABLESPACE <TS-NAME>;  
  
RMAN> BACKUP DATAFILE <DF-NAME>;  
  
RMAN> BACKUP RECOVERY AREA;  
  
RMAN> BACKUP CURRENT CONTROLFILE;  
  
RMAN> BACKUP SPFILE;  
  
RMAN> BACKUP NOT BACKED UP SINCE TIME 'SYSDATE - 7'  
DATABASE;  
  
RMAN> BACKUP DATABASE AS COPY;
```

- Backup mit RMAN wenn im „NOARCHIVELOG“ Modus

```
rman / AS SYSDBA  
RMAN> SHUTDOWN IMMEDIATE;  
RMAN> STARTUP MOUNT;  
RMAN> BACKUP DATABASE;  
RMAN> ALTER DATABASE OPEN;
```

- Backup mit RMAN wenn im „ARCHIVELOG“ Modus

```
rman / AS SYSDBA
```

```
RMAN> BACKUP DATABASE;
```

- Besser gleich mit den archivierten Redolog-Dateien

```
rman / AS SYSDBA
```

```
RMAN> BACKUP DATABASE PLUS ARCHIVELOG;
```

- Benutzerfehler
- Anweisungsfehler
- Prozessfehler
- Netzwerkfehler
- Instancefehler
- Medienfehler

- **Reaktion je nach Fehlertyp**
- **evtl. Wiederherstellung des alten Datenbestands**
  - Import eines zuvor durchgeführten Datenexports
  - Unvollständige Wiederherstellung
    - Aufwändig, da Oracle/RMAN kein Object-Level Recovery bietet
  - Flashback-Techniken, z.B. Flashback Query
- **Überprüfung der Sicherheits-Mechanismen**

- **Statement-Level-Rollback (automatisch)**
- **Analyse der Fehlerursache**

- Oracle-Server-Prozess (dedicated oder multi-threaded) stürzt ab
- PMON-Prozess führt automatisch ein Rollback durch und gibt Ressourcen / Sperren wieder frei

- Falls nur ein Serverprozess beteiligt: PMON-Prozess führt automatisch Recovery durch (wie Prozessfehler)
- Spezialfall: PC "ausschalten": Erkennung durch Dead Connection Detection
- Verteilte Transaktionen mit Two-Phase-Commit: automatisches Recovery durch RECO-Prozess

- Verursacht durch Hardware-, Software-, oder Bedienfehler
- Automatisches Instance Recovery beim nächsten Neustart
- Eventuell vorher SHUTDOWN ABORT notwendig
- Analyse der Fehlerursache
  - Insbesondere Alert-Log und Trace-Dateien

- **Physikalischer Defekt einer Platte**
- **Recovery-Aktion des DB-Administrators notwendig**
- **Palette der Möglichkeiten abhängig von der DB-Vorbereitung**
  - Archivelog- vs Noarchivelog-Modus
  - Flashback Logging aktiv
  - Vorhandene Backups / Backup-Strategie
  - Standby-Datenbanken

- **ARCHIVELOG-Modus**
  - Online Redologs und archivierte Redologs verfügbar
  - Media Recovery möglich
- **NOARCHIVELOG-Modus**
  - nur Online Redologs verfügbar
  - Kein Media Recovery möglich, nur Instance Recovery

- **Vollständiges Recovery (alle Transaktionen werden wiederhergestellt) NUR DANN möglich, wenn**
  - Kontrolldatei aktuellen Zustand hat (KEIN Restore!),
  - notwendige Backups der Datendateien vorliegen und
  - lückenlose Folge von archivierten/Online Redologs bis zum aktuellen Zeitpunkt vorhanden.
- **Ansonsten unvollständiges Recovery!**

- **Alle Dateien der Datenbank wiederherstellen:**
  - Datafiles
  - Controlfiles
  - Redo Log-Files
  - Password-File (optional)
  - spfile<SID>.ora bzw. init<sid>.ora (optional)
- **Vorteile:**
  - Leicht zu handhaben, je nach Datenmenge relativ geringe Zeit zur Wiederaufnahme des Betriebs
- **Nachteile:**
  - Alle Daten, die seit dem letzten Backup erfasst wurden, gehen verloren

- **Nur beschädigte Dateien wiederherstellen**
- **Vorteile:**
  - Nur beschädigte Dateien müssen wiederhergestellt werden
  - Alle Daten bleiben erhalten
- **Nachteile:**
  - Archivierte Redolog-Dateien müssen seit dem letzten Backup bis zum Ausfallzeitpunkt vorhanden sein

# Was passiert beim „Wiederherstellen“

1. Datendateien werden zurückgesichert → Restore
2. Transaktionen werden wieder hergestellt → Recovery
3. Datenbank wird geöffnet / Steht wieder vollständig zur Verfügung

- **Closed-Database-Recovery:**
  - Wiederherzustellende Dateien gehören
    - zum System-Tablespace oder
    - zu Tablespaces mit Undo-Segmenten (normalerweise nur der Undo-Tablespace)
  - Die gesamte Datenbank ist im MOUNT-Status  
→ für Anwender nicht verfügbar
  - Die gesamte Datenbank bzw. der größte Teil der Dateien muss wiederhergestellt werden
- **Opened-Database-Recovery:**
  - Wiederherzustellenden Dateien gehören zu anderen Tablespaces
  - Die Datenbank kann geöffnet bleiben. Die betroffenen Datendateien sind offline  
→ Auf den restlichen Daten kann weitergearbeitet werden

- **V\$RECOVER\_FILE** enthält alle Datenfiles, die wiederhergestellt werden müssen
- **V\$DATAFILE** enthält alle Datenfiles. Spalte **STATUS** gibt an, ob das Datafile Recovery benötigt
- **V\$RECOVERY\_LOG** enthält alle archivierte Redologs, die zum Recovery benötigt werden
- **V\$LOG\_HISTORY** enthält alle archivierten Redologs der Datenbank

- **Verlust eines Tablespaces**

```
rman target /  
RMAN> SQL 'ALTER TABLESPACE users OFFLINE IMMEDIATE';  
RMAN> RESTORE TABLESPACE users;  
RMAN> RECOVER TABLESPACE users;  
RMAN> SQL 'ALTER TABLESPACE users ONLINE';
```

- **Verlust einer Datendatei**

```
SQL> ALTER DATABASE DATAFILE '/oradata/JA10G/users01.dbf' OFFLINE IMMEDIATE;  
rman target /  
RMAN> RESTORE DATAFILE '/oradata/JA11G/users01.dbf';  
RMAN> RECOVER TABLESPACE '/oradata/JA11G/users01.dbf';  
SQL> ALTER DATABASE DATAFILE '/oradata/JA11G/users01.dbf' ONLINE;
```

- **Verlust der ganzen Datenbank**

```
sqlplus system/manager@SUNDB as sysdba
SQL> STARTUP NOMOUNT;
SQL> EXIT;
rman
RMAN> SET DBID=2749674549
RMAN> CONNECT TARGET SYSTEM/MANAGER@SUNDB
RMAN> SET CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO
'/oradata2/backup/%F.ctl';
RMAN> RESTORE CONTROLFILE FROM AUTOBACKUP;
RMAN> ALTER DATABASE MOUNT;
RMAN> RESTORE DATABASE;
RMAN> RECOVER DATABASE;
RMAN> ALTER DATABASE OPEN RESETLOGS;
```

- **Verlust von Online Redologs**
- **Falls nur Log-Member (Spiegel) verloren: Spiegel wieder aufsetzen**
- **Schlimmster Fall: Verlust einer kompletten Log-Gruppe**
  - SHUTDOWN ABORT
  - Restore der kompletten DB
  - Unvollständiges Recovery!

- Verlust von archivierten Redologs
- Kein unmittelbares Problem
- **Aber: Recovery-Lücke bei zusätzlicher Beschädigung einer Datendatei!**
  - Backup so schnell wie möglich!
- **Bei zusätzlicher Beschädigung einer Datendatei**
  - Evtl. nur unvollständiges Recovery möglich!

- **Verlust der Kontrolldateien**

- Controlfile-Backup einspielen oder
- mit `CREATE CONTROLFILE` neu erzeugen
  - Umständlich: alle Daten- und Redolog-Dateien angeben!
  - Einfacher:  
`CREATE CONTROLFILE` -Kommando präventiv erzeugen durch:  
`ALTER DATABASE BACKUP CONTROLFILE TO TRACE;`

- **Gründe**

- Der Versuch, ein vollständiges Recovery durchzuführen schlug fehl, da eine oder mehrere archivierte Redologs fehlen
- Alle Controlfiles sind unbrauchbar
- Alle Online Redologs und ein Datafile sind unbrauchbar
- Benutzerfehler
  - eine wichtige Tabelle wurde gelöscht
  - wichtige Tabelleninhalte wurden gelöscht
  - es wurden fehlerhafte Daten in eine Tabelle eingetragen

- **Techniken**

- Zurücksetzen über Restore / Recovery
  - Restore eines älteren Backups
  - Unvollständiges Recovery bis Zeitpunkt oder SCN
- Zurücksetzen über Flashback Database

# Unvollständiges Recovery - Modi

- **Bis zu einer Redolog-Nummer**

```
RMAN> RECOVER DATABASE UNTIL SEQUENCE 987;
```

- **Bis zu einem Zeitpunkt**

```
RMAN> RECOVER DATABASE UNTIL TIME  
      '2000-02-25 23:00:00';
```

- **Voraussetzungen**

- Flash-Recovery-Area ist gesetzt

```
SQL> ALTER SYSTEM SET db_file_recovery_dest = '/oraflash';
```

```
SQL> ALTER SYSTEM SET db_file_recovery_dest_size = 10G;
```

- Flashback Retention Target (in Minuten) ist gesetzt

```
SQL> ALTER SYSTEM SET db_flashback_retention_target = 1440;
```

- Flashback-Logging ist eingeschaltet

```
SQL> SHUTDOWN IMMEDIATE;
```

```
SQL> STARTUP MOUNT;
```

```
SQL> ALTER DATABASE FLASHBACK ON;
```

```
SQL> ALTER DATABASE OPEN;
```

- **Herunterfahren der Datenbank und Zurücksetzen der Datenbank auf eine bestimmte Uhrzeit**

```
SQL> SHUTDOWN IMMEDIATE;  
  
SQL> STARTUP MOUNT;  
  
SQL> FLASHBACK DATABASE TO TIMESTAMP  
    TO_DATE('11.08.2007 13:54:16', 'DD.MM.YYYY HH24:MI:SS');
```

- **Überprüfung, ob der Stand der Datenbank richtig ist**

```
SQL> ALTER DATABASE OPEN READ ONLY;
```

- **Bei erfolgreichem Recovery Datenbank öffnen**

```
SQL> SHUTDOWN IMMEDIATE  
  
SQL> STARTUP;
```

- **Verringerung der Downtime durch:**
  - Parallelisieren des Recovery
    - Parallelisierung des Recovery kann durch den Serverparameter `RECOVERY_PARALLELISM` bestimmt werden
    - Parameterwert gibt den Grad der Parallelisierung an
    - Optimal: ein bis zwei Prozesse pro Festplatte, die Datafiles enthält
  - Hochfahren der Datenbank und Begrenzung des Recovery auf fehlerhafte Tablespaces
  - Neuanlegen nicht erforderlicher bzw. rekonstruierbarer Tablespaces, anstatt diese zu recovern

- **Alert-Datei (alert<SID>.log) im Background-Dump-Dest**
  - Informationen und Fehler, die im laufenden Betrieb erzeugt werden
  - den aktuellen Status des Systems
  - neueste Informationen immer am Ende der Datei
  - Datum und Uhrzeit von wichtigen Systemabläufen
- **DBVERIFY:**
  - Programm, um Datafiles auf korrupte Blöcke zu überprüfen
- **ANALYZE VALIDATE STRUCTURE:**
  - SQL-Anweisung, um die Integrität einer Tabelle oder eines Index zu überprüfen
- **DBMS\_REPAIR:**
  - PL/SQL-Package, zur Erkennung und Markierung von korrupten Blöcken

- **Data Guard / Standby Database**
  - Kopie der Produktionsdatenbank und Recovery der Redolog-Informationen
- **Flashback Table**
  - Wiederherstellen einer Tabelle nach dem Befehl „DROP TABLE“ → Recyclebin
  - Zurücksetzen einer Tabelle auf einen älteren Zustand über die UNDO-Informationen
- **LogMiner**
  - Analysetool für die Redolog-Dateien
  - Fehlerhafte Transaktionen können ggf. rückgängig gemacht werden

- **Welche Voraussetzung muss für einen Online Backup der Oracle Datenbank mindestens erfüllt sein?**
  1. Es darf keine offenen Transaktionen geben
  2. Der Archivelog-Modus muss eingeschaltet sein
  3. Es dürfen nur Abfragen gemacht werden
  4. Der Flashback-Modus muss eingeschaltet sein

- Welche Teile der Datenbank müssen bei einem Online Backup NICHT gesichert werden?
  1. UNDO Tablespace
  2. Logbuffer
  3. Archivierte Redolog-Dateien
  4. SYSTEM Tablespace

- **Es wird täglich eine Vollsicherung der Datenbank durchgeführt und stündlich eine Sicherung der archivierten Redolog-Dateien. Was sagt das über die RPO?**
  1. Die RPO beträgt 99,99%
  2. Es können Daten der letzten Stunde verloren gehen
  3. Es dauert 60 Minuten, bis die Datenbank wieder zur Verfügung steht
  4. Die RPO bezieht sich nicht auf die Sicherung sondern auf die Größe der Datenbank