



## Flashback – Reise in die Vergangenheit

**Warum Oracle Zeitreisen anbieten kann,  
der Microsoft SQL Server aber leider nicht.**

PASS Regionaltreffen  
15.04.2015, Berlin

[Andreas.Jordan@ordix.de](mailto:Andreas.Jordan@ordix.de)

- Werbung
- Ein erster Eindruck
- Ablauf einer Transaktion
- Lesekonsistenz
- Flashback
- „Flashback“ mit dem MS SQL Server

*Werbung*

- Seit über 20 Jahre in der IT-Branche tätig
- Seit über 12 Jahren als Consultant bei der ORDIX AG
- Microsoft:
  - Microsoft Certified Solutions Associate (MCSA): SQL Server 2012
  - Microsoft Certified Solutions Expert (MCSE): Data Platform
  - Microsoft Certified Trainer (MCT)
- Und darüber hinaus:
  - Oracle PL/SQL-Entwicklung und -Optimierung, Datenbankadministration
  - Windows, Unix
  - VBA, VB.net, Perl, Python, Shell
  - Nagios

- Seit 24 Jahren am Markt mit:

- Beratung
- Entwicklung
- Service
- Training
- Projektmanagement

- Microsoft:

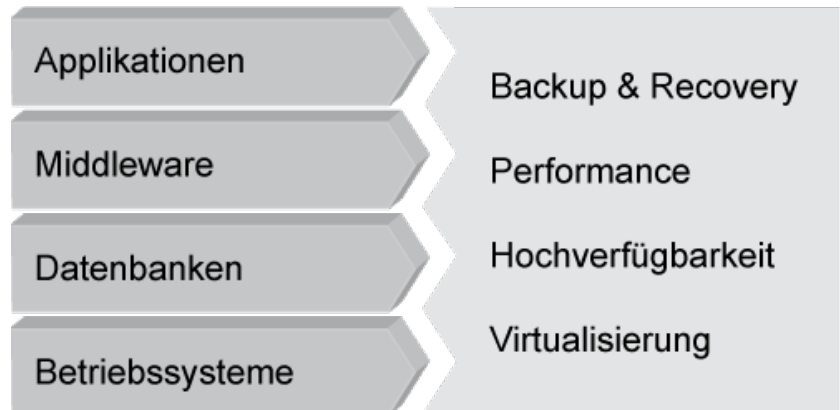
- Silver Data Platform Partner

- 5 Standorte

- Paderborn (Zentrale)
- Wiesbaden (Seminarzentrum)
- Münster, Köln, Gersthofen

- 111 Mitarbeiter

- 14,7 Mio. € Umsatz



*Ein erster Eindruck*

# Wie sahen die Daten gestern aus?



```
SELECT * FROM mitarbeiter AS OF TIMESTAMP SYSDATE - 1;
```

## *Ablauf einer Transaktion*



- Eine Transaktion bedeutet immer „Alles oder Nichts“
- Wichtig dabei: Der Weg zurück muss gesichert werden
- Zu jeder Datenänderung wird also die entgegen gesetzte Anweisung generiert und gesichert

Mitarbeiter	
MaNr	Gehalt
0815	3000
1234	8000
4711	<del>8000</del> 6000

Redo-Information:

```
UPDATE Mitarbeiter  
SET Gehalt = 6000  
WHERE MaNr = 4711;
```

Undo-Information:

```
UPDATE Mitarbeiter  
SET Gehalt = 5000  
WHERE MaNr = 4711;
```



- Nach der Bestätigung des COMMIT verlässt sich der Nutzer auf die Gültigkeit der Änderungen
- Da die eigentlichen Daten in den Datendateien erst später aktualisiert werden, müssen zumindest die Änderungsanweisungen gespeichert sein

Mitarbeiter	
MaNr	Gehalt
0815	3000
1234	8000
4711	<del>8000</del> 6000

Redo-Information:

```
UPDATE Mitarbeiter  
SET Gehalt = 6000  
WHERE MaNr = 4711;
```



Undo-Information:

```
UPDATE Mitarbeiter  
SET Gehalt = 5000  
WHERE MaNr = 4711;
```



- Im Hauptspeicher: Log Buffer / Log Cache
  - Relativ kleiner Bereich (wenige MB)
  - Enthält die Redo- und die Undo-Informationen als Statements
  - Wird in kurzen Intervallen, spätestens beim Commit auf Festplatte gesichert
- Auf der Festplatte im direkten Zugriff: Transaktionsprotokoll
  - Kann relativ groß werden (mehrere GB)
  - Wird ins Backup übertragen, wenn die Transaktion abgeschlossen ist
- Auf der Festplatte zur Sicherung: Transaktionsprotokoll-Backup
  - Zur Wiederherstellung der Datenbank

- Im Hauptspeicher: Redo Log Buffer
  - Relativ kleiner Bereich (wenige MB)
  - Enthält die Redo- und die Undo-Informationen als Statements
  - Wird in kurzen Intervallen, spätestens beim Commit auf Festplatte gesichert
- Im Hauptspeicher und auf Festplatte im direkten Zugriff: Undo Tablespace
  - Enthält die Daten-Werte aus den Undo-Informationen
  - Wird auch über das Ende der Transaktion hinaus gespeichert
  - Wird (auf Festplatte) auch über den Neustart der Instanz hinaus gespeichert

- Auf der Festplatte im rein schreibenden Zugriff: Redo Log Files
  - Mindestens zwei Dateien mit einer festen Größe (min. 4 MB)
  - Wenn eine Datei voll ist, wird diese abgeschlossen und die andere verwendet
  - Abgeschlossene Dateien werden vom Archiver in die Archive Destination kopiert und zur nächsten Verwendung freigegeben
  
- Auf der Festplatte zur Sicherung: Archived Redo Log Files
  - Zur Wiederherstellung der Datenbank
  - Werden regelmäßig aus der Archive Destination auf ein Backup-Medium übertragen

- Gemeinsamkeiten
  - Drei Ebenen: Hauptspeicher / Festplatte im Zugriff / Festplatte ohne Zugriff
- MS SQL Server
  - Undo-Daten werden relativ schnell aus dem Zugriff entfernt (Backup)
  - Undo-Daten werden nur zur Absicherung der aktuellen Transaktion genutzt
- Oracle
  - Undo-Daten werden (längerfristig und persistent) in eigenem Bereich gespeichert
  - Undo-Daten werden auch zur Sicherstellung der Lesekonsistenz genutzt

## *Lesekonsistenz*

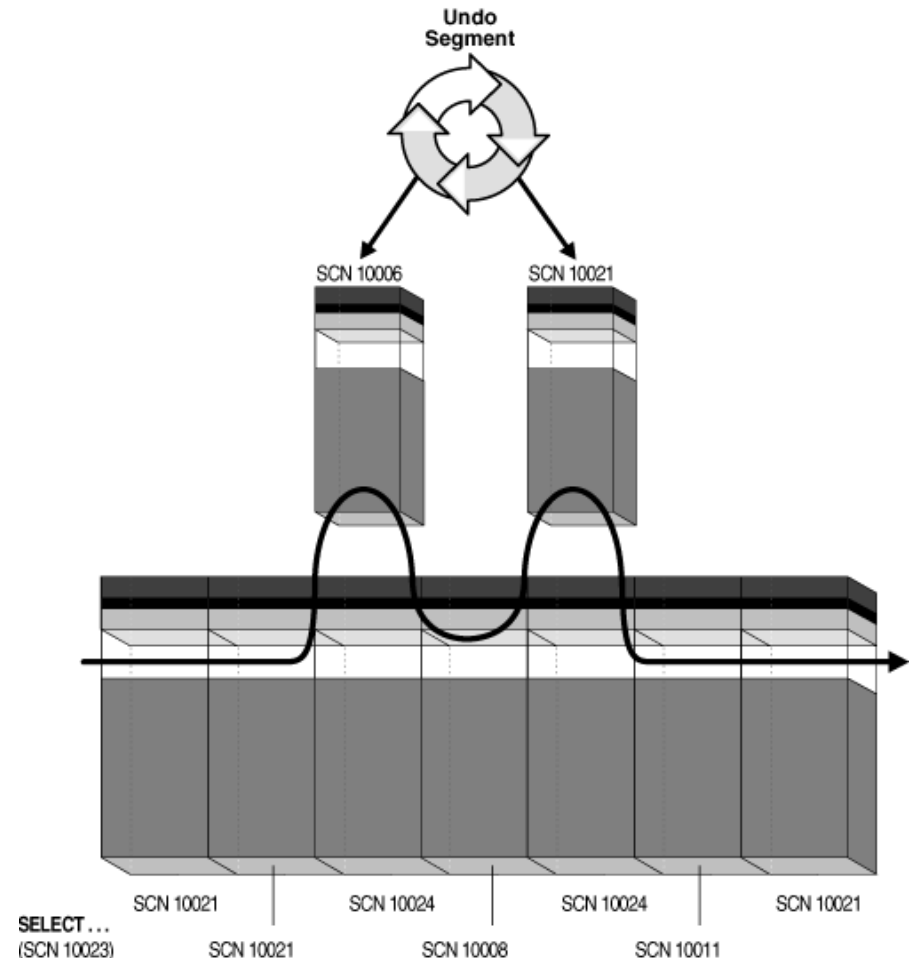
*„Die Lesekonsistenz stellt sicher, dass der Datenbankennutzer auch bei langanhaltenden Transaktionen auf einen konsistenten Datenbankzustand zugreifen kann.“*

[http://wikis.gm.fh-koeln.de/wiki\\_db/Datenbanken/Transaktion,Lesekonsistenz](http://wikis.gm.fh-koeln.de/wiki_db/Datenbanken/Transaktion,Lesekonsistenz)



- Standard: Isolationslevel „Read Committed“
  - Ändernde Session erzeugt eine Sperre auf veränderte Datensätze
  - Lesende Zugriffe werden durch diese Sperre blockiert
- Alternative: Isolationslevel „Snapshot“ oder „Read Committed Snapshot“
  - Versionen von Zeilen werden in tempdb gespeichert
  - Benötigt 14 Bytes pro Datensatz
  - Lesende Zugriffe werden nicht blockiert sondern nutzen Zeilen-Versionen
- Wie lange sind die Zeilen-Versionen verfügbar?
  - Nur bis zum Neustart der Instanz, da die tempdb beim Start neu erstellt wird

- Standard: Isolationslevel „Read Committed“
  - Lesende Zugriffe werden nicht blockiert sondern nutzen Undo-Informationen
  - Vorherige Versionen werden bei Bedarf erzeugt
  - Risiko: Undo-Information ist nicht mehr verfügbar, Abfrage bricht dann ab



## *Flashback*

- Lesekonsistenz:
  - Rekonstruktion von Informationen zum Zeitpunkt des Beginns der Abfrage
  - Basis sind die Undo-Informationen aus dem Undo Tablespace
- Flashback Query:
  - Rekonstruktion von Informationen zu einem beliebigen Zeitpunkt
  - Die benötigten Undo-Informationen müssen allerdings noch vorhanden sein

- **UNDO\_RETENTION**
  - Option für das System
  - Angabe in Sekunden, wie lange die Undo-Daten mindestens nach Ende der Transaktion noch vorgehalten werden sollen
  - Wird nur für automatisch vergrößernde Undo Tablespaces beachtet und auch nur, solange MAXSIZE noch nicht erreicht ist
  
- **RETENTION GUARANTEE**
  - Option für den Undo Tablespace
  - Garantiert die Einhaltung der UNDO\_RETENTION
  - Kann zu Rollbacks von Transaktionen führen, wenn nicht mehr genug Platz vorhanden ist

- Flashback Query:
  - `SELECT * FROM mitarbeiter AS OF TIMESTAMP SYSDATE - 1;`
  - `SELECT * FROM mitarbeiter AS OF SCN 12345;`
- Flashback Query Versions Between
  - `SELECT * FROM mitarbeiter VERSIONS BETWEEN ... AND ...;`
- Flashback Table:
  - `FLASHBACK TABLE mitarbeiter TO ...;`
- Flashback Table Drop:
  - `FLASHBACK TABLE mitarbeiter TO BEFORE DROP;`

- Flashback Database:
  - FLASHBACK DATABASE TO ...
  - Point-in-Time-Recovery ohne Backup
  - Nutzt die Flash Recovery Area
- Flashback Data Archive:
  - Archivierung der Veränderungen an einzelnen Tabellen für längere Zeit
- Flashback Transaction Backout:
  - Rollback von Transaktionen

- Auch andere Datenbankmanagementsysteme haben schöne Features
- „Read Committed“ kann mit unterschiedlichem Sperrverhalten implementiert sein
- Die getrennte und persistente Speicherung der Undo-Daten ist der Schlüssel zur Zeitreise
- Verwendung von Flashback in Produktionsumgebungen eher selten und fraglich, aber ideal für Entwicklung und Test



- ORDIX-Schulung „Oracle Datenbankadministration Aufbau“  
<http://training.ordix.de/siteengine/action/load/nr/61/index.html>
- Oracle Doku: Managing Undo  
[http://docs.oracle.com/cd/E11882\\_01/server.112/e25494/undo.htm](http://docs.oracle.com/cd/E11882_01/server.112/e25494/undo.htm)
- Oracle Doku: Using Oracle Flashback Technology  
[http://docs.oracle.com/cd/E11882\\_01/appdev.112/e41502/adfns\\_flashback.htm](http://docs.oracle.com/cd/E11882_01/appdev.112/e41502/adfns_flashback.htm)

*„Flashback“ mit dem MS SQL Server*

- Auch beim MS SQL Server liegen alle benötigten Informationen vor, sind aber evtl. nicht mehr im Zugriff.
  - Aktives Transaktionsprotokoll
  - Gesichertes Transaktionsprotokoll (Backup)
  - Nicht angefügte Datenbank (nicht aktive LDF-Datei)

- Hier hilft die Software „ApexSQL Log“:  
[http://www.apexsql.com/sql\\_tools\\_log.aspx](http://www.apexsql.com/sql_tools_log.aspx)

ApexSQL Log

## Discovery and recovery tool

Explore the SQL transaction log and undo transactions  
Audit schema and data changes

## ApexSQL Log

## Discovery and recovery tool

Explore the SQL transaction log and undo transactions  
Audit schema and data changes

Download trial

Overview

Features

Testimonials


Content


"...as an insurance policy against accidental SQL data deletion, etc ... I am very pleased to say that the ApexSQL Log Tool has exceeded our expectations and I now consider it a necessity when working with SQL databases"


**Craig Hatley**, Director of Operations, Iatric Systems, Inc.


[See more](#)


Home
View
Resources



New



Open



Session settings



Save


Print


Print preview


Export


Create undo script


Create redo script

Sessions
Print
Actions

Operation	Schema	Object
<input checked="" type="checkbox"/> UPDATE	dbo	Customers
<input type="checkbox"/> INSERT	dbo	AUDIT_LOG_TRANSACTIONS
<input type="checkbox"/> INSERT	dbo	AUDIT_LOG_DATA
<input checked="" type="checkbox"/> UPDATE	dbo	Customers

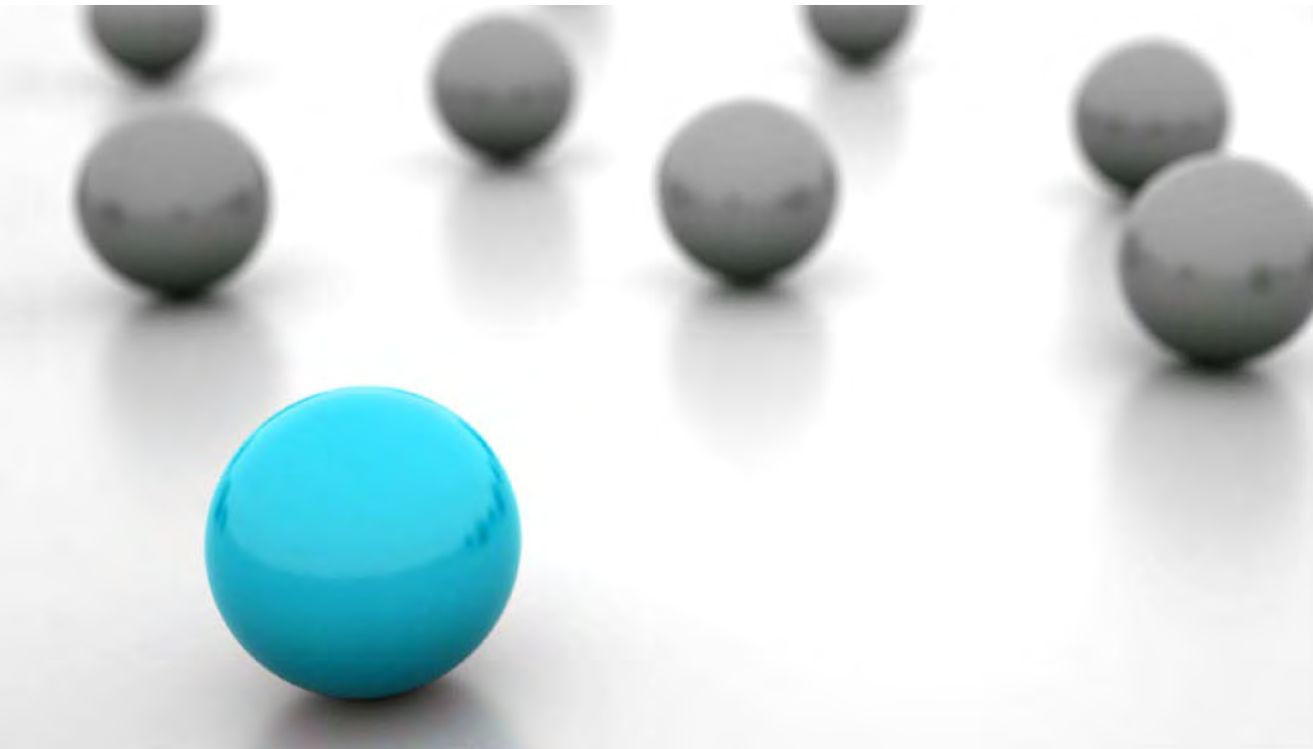
Order now

\$1,599

Buy online

or

+1 (866) 665-5500  
sales@apexsql.com



Zentrale Paderborn  
Westernmauer 12 - 16  
33098 Paderborn  
Tel.: 05251 1063-0

Seminarzentrum Wiesbaden  
Kreuzberger Ring 13  
65205 Wiesbaden  
Tel.: 0611 77840-00

Zentrales Fax:  
0180 1 67349 0  
0180 1 ORDIX 0

Weitere Geschäftsstellen  
in Köln, Münster und Gersthofen

E-Mail: [info@ordix.de](mailto:info@ordix.de)  
Internet: <http://www.ordix.de>

## Vielen Dank für Ihre Aufmerksamkeit!