



# Flashback

---

## Früher war alles besser

**Marion Mahr**  
Senior Consultant

**Daniel Schulz**  
Solution Architect

OPITZ CONSULTING GmbH



München, 14.09.2012



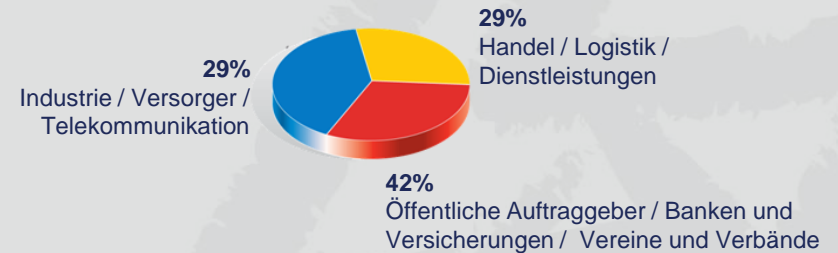
## Mission

Wir entwickeln gemeinsam mit allen Branchen Lösungen, die dazu führen, dass sich diese Organisationen besser entwickeln als ihr Wettbewerb.

Unsere Dienstleistung erfolgt partnerschaftlich und ist auf eine langjährige Zusammenarbeit angelegt.

## Märkte

- Branchenübergreifend
- Über 600 Kunden



## Leistungsangebot

- Business IT Alignment
- Business Information Management
- Business Process Management
- Anwendungsentwicklung
- SOA und System-Integration
- IT-Infrastruktur-Management

## Eckdaten

- Gründung 1990
- 400 Mitarbeiter
- 8 Standorte



# Agenda

---

- 1. Einleitung**
- 2. Flashback Query**
- 3. Flashback Table**
- 4. Flashback Drop (Recycle Bin)**
- 5. Flashback Database**
- 6. Flashback Data Archive (Total Recall)**

1

# Einleitung

# Warum die Vergangenheit ansehen?

---

## Gesetzliche Vorgaben (Sarbanes Oxley oder Basel-II)

### Typische Fehler

- Update mit falscher oder ohne(!) WHERE-Klausel
- Tabelle gelöscht oder mit TRUNCATE geleert
- Benutzer gelöscht

→ **Recoverybedarf entsteht überwiegend durch menschliche Fehler**

# Der klassische Ansatz nach Fehlern

---

- **Zeitpunkt ermitteln falls nicht bekannt (z.B Logminer)**
- **Point-In-Time-Recovery der Datenbank bis kurz vor den Fehler**
- **Sehr langwierig bei großen Datenbanken**
- **Wenn Zeitpunkt überschritten wird, muss der Vorgang wiederholt werden**

# Flashback hilft

---

**Viele typische Fehlerszenarien können ohne Recovery gelöst werden!**

**Vereinfachte Ermittlung des Fehlerzeitpunktes**

**Verschiedene Funktionalitäten, um gelöschte Daten wiederherzustellen oder ältere Versionen zu betrachten**

**Zurückschreiben auf Datenbank-, Tabellen- und Transaktionsebene**

**Schneller als klassisches Point-In-Time-Recovery**

# Flashback in Versionen und Editionen

---

## ■ In Standard Edition:

- Seit 9i: Flashback (Version) Query
- Ab 10g: Flashback Drop (Recycle Bin)

## ■ Nur Enterprise Edition:

- Ab 10g:
  - Flashback Table
  - Flashback Database
  - Flashback Transaction Query
- Neu in 11g:
  - Flashback Transaction Backout
  - Flashback Data Archive – kostenpflichtige Option!

## ■ Trotz des gemeinsamen Namens sind hier unterschiedliche Techniken in Verwendung!

## ■ Die Flash Recovery Area hat auch nur bedingt damit zu tun!



# Terminologie

---

## ■ **Flashback Query**

Tabelleninhalt zu früherem Zeitpunkt ansehen

## ■ **Flashback Drop**

Gelöschte Tabelle wiederherstellen

## ■ **Flashback Table**

Tabelleninhalt zu einem bestimmten Zeitpunkt wiederherstellen

## ■ **Flashback Database**

Point-In-Time-Recovery „rückwärts“

## ■ **Flashback Transaction (Backout)**

Bestimmte Transaktion rückgängig machen

## ■ **Flashback Data Archive**

Historisierung von Änderungen über Jahre

# 2

## Flashback Query

# Flashback Query für

---

## Antworten auf folgende Fragen

- **Wie sahen die Daten heute morgen aus?**
- **Welche Veränderungen sind heute morgen passiert?**
- **Wann genau wurde das falsche SQL ausgeführt?**
- **Wie können die Änderungen rückgängig gemacht werden?**
- **Wie kam es überhaupt zu diesem Fehler?**

## Verfahren: Auslesen des UNDO-Inhaltes

# Flashback Version/Transaktion/Query aktivieren

---

- **Automatic Undo Management muss verwendet werden**
  - UNDO\_MANAGEMENT = AUTO
- **UNDO Retention anpassen**
  - Voreinstellung sind 900 sek (= 15 min)
- **Bei Abfragen über diese Zeit hinaus kann es zum Fehler ORA-30052 kommen**
- **Bei akutem Platzmangel werden auch vor Ablauf der UNDO Retention die ältesten Daten über abgeschlossene Transaktionen überschrieben**
- **Bei Bedarf UNDO auf Retention Garantie stellen**
  - Achtung: ist UNDO TBS zu klein, blockiert die Instanz wie bei Archiver Stuck!

# Flashback (Version) Query

---

```
SELECT <Spalten>  
FROM <Tabelle>  
AS OF TIMESTAMP <timestamp>;
```

```
SELECT <Spalten>  
FROM <Tabelle>  
VERSIONS BETWEEN TIMESTAMP <start_timestamp>  
                AND <end_timestamp>;
```

```
SELECT <Spalten>  
FROM <Tabelle>  
VERSIONS BETWEEN TIMESTAMP minvalue  
                AND maxvalue;
```

■ **SCN kann ebenfalls verwendet werden!**

# Beispiel

---

- **Benutzer Meier meldet, Abteilung 50 ist „verschwunden“.**
- **Er ist sehr aufgebracht und möchte wissen, was passiert ist.**
- **Morgens um 7 Uhr war noch alles in Ordnung.**
- **Wieso ist die Datenbank jetzt defekt?**
- **Kann doch nicht sein!**
- **Unprofessionelle DBAs!**

# Flashback Query Beispiel

```
SELECT * FROM dept
AS OF TIMESTAMP
to_timestamp('12-05-2009 07:00:00', 'dd-mm-yyyy hh24:mi:ss');
```

DEPTNO	DNAME	LOC
-----	-----	-----
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON
50	IT	FRANKFURT

# Flashback Version Query Beispiel

```
SELECT deptno, dname, loc, versions_operation, versions_xid, versions_starttime
FROM scott.dept
VERSIONS between timestamp minvalue and maxvalue
ORDER BY deptno, versions_starttime;
```

D	DNAME	LOC	V	VERSIONS_XID	VERSIONS_STARTTIME
--	-----	-----	-	-----	-----
10	ACCOUNTING	NEW YORK			
20	RESEARCH	DALLAS			
30	SALES	CHICAGO			
40	OPERATIONS	BOSTON			
50	IT	FRANKFURT			
50	MARKETING	FRANKFURT	U	1E000A0008000000	12.05.09 09:30:02
50		FRANKFURT	U	1E000B0008000000	12.05.09 09:31:15
50		FRANKFURT	D	1E000C0008000000	12.05.09 09:32:20



# Wer war es denn nun?

## ■ Verbinden der Transaktion mit Anwender:

```
SQL> SELECT logon_user, table_name, table_owner, undo_sql
FROM flashback_transaction_query
WHERE table_owner='SCOTT'
AND xid= '1E000C0008000000';
```

LOGON_USER	TABLE_NAME	TABLE_OWNER	UNDO_SQL
-----	-----	-----	-----
MEIER	DEPT	SCOTT	insert into "SCOTT"."DEPT" ("DEPTNO", "DNAME", "LOC") values ('50', ' , 'Frankfurt');

# UNDO SQL für mehrere Transaktionen

```
SQL> SELECT undo_sql FROM flashback_transaction_query
WHERE table_owner = 'SCOTT' AND table_name = 'DEPT'
ORDER BY commit_timestamp desc;
```

```
UNDO_SQL
```

```
-----
insert into "SCOTT"."DEPT" ("DEPTNO", "DNAME", "LOC")
  values ('50', '', 'Frankfurt');
update "SCOTT"."DEPT" set "DNAME" = 'MARKETING' where
  ROWID = 'AAALy6AAEAAAAAOAAA';
update "SCOTT"."DEPT" set "DNAME" = 'IT' where ROWID =
  'AAALy6AAEAAAAAOAAA';
```

# Flashback Version Backout

- **Neu in 11g, Integriert zusätzlich den Logminer**

- **Benötigt Supplemental Log Data**

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;  
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA  
    (PRIMARY KEY) COLUMNS;  
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA  
    (FOREIGN KEY) COLUMNS; (11.2)  
dbms_flashback.transaction_blackout..
```

- **Kann Abhängigkeiten behandeln, z.B. UPDATE auf einer zuvor eingefügte Zeile (WAW – Write after Write):**

- **Default:** Nur konfliktfreie Transaktionen zurücknehmen
- **NOCASCADE FORCE:** Nur diese eine Transaktion, auch wenn es Abhängigkeiten gibt
- **CASCADE:** Alle abhängigen Transaktionen zurückrollen

# 3

## Flashback Table

# Flashback Table

---

- Tabelleninhalt wird zu einem bestimmten Zeitpunkt wiederhergestellt
- Basiert auf UNDO Informationen
- Row Movement muss aktiviert sein
- Zugehörige Tabellen haben exklusives DML Lock während der Ausführung
- Abhängige Tabellen können gleichzeitig behandelt werden

```
SQL> FLASHBACK TABLE employees, departments TO TIMESTAMP  
TO_TIMESTAMP ('13-May-2009,10:00', 'dd-mon-yyyy, hh24:mi');
```

# Einschränkungen Flashback Table

---

- **Keine Systemtabellen**
- **Nicht über DDL Anweisungen auf der Tabelle hinweg**
- **Nicht über SHRINK hinweg**
- **Statistiken werden nicht zurückgesetzt!**
- **Zwischenzeitlich gelöschte Indizes werden nicht wiederhergestellt**
- **Werden Constraints verletzt, so bricht das Flashback ab**

# 4

## Flashback Drop (Recycle Bin)

# Flashback Drop (Recycle Bin)

- Ab 10g
- Objekte werden nicht sofort komplett aus dem System entfernt
- Gelöschte Objekte werden nur im Data Dictionary umbenannt (BIN\$...) – nicht verschoben
- Ist als Voreinstellung aktiv, deaktivieren durch:

```
ALTER SYSTEM SET "_recyclebin"=FALSE SCOPE=BOTH; -- 10gR1
```

```
ALTER SYSTEM SET recyclebin=off; -- ab 10gR2
```

- **SELECT** weiterhin möglich, wenn der neue Tabellenname angegeben wird:

```
SELECT * FROM "BIN$jYKhVPTQlyDjXynS8Noxw==$0";
```



# Recycle Bin anzeigen

```
SQL> show recyclebin
```

ORIGINAL NAME	RECYCLEBIN NAME	OBJECT TYPE	DROP TIME
TEST	BIN\$jTYKhVPTQlyDjXynS8Noxw==\$0	TABLE	2005-12-04:22:21:53

```
SQL> select object_name,original_name from user_recyclebin;
```

OBJECT_NAME	ORIGINAL_NAME
BIN\$bwFy+Ua2QHynpfphUWZYWw==\$0	JHIST_JOB_IX
BIN\$vU5OgwqdT1G5jVOFNuh/QQ==\$0	JHIST_EMPLOYEE_IX
BIN\$gKvQUsiYS3CHnSA+d1/4Tw==\$0	JHIST_DEPARTMENT_IX
BIN\$Z6ufu4NYSxObwLpUPY90Fw==\$0	JHIST_EMP_ID_ST_DATE_PK
BIN\$rBX05CwKR9CYgznjIZkkjg==\$0	JOB_HISTORYb

# Objekte aus Recycle Bin zurückholen

## ■ Tabelle wiederherstellen (inkl. abhängige Objekte)

```
FLASHBACK TABLE JOB_HISTORY TO BEFORE DROP;  
FLASHBACK TABLE 'BIN$rBX05CwKR9CYgznjIZkkjg==$0' TO  
BEFORE DROP;  
FLASHBACK TABLE 'BIN$rBX05CwKR9CYgznjIZkkjg==$0' TO  
BEFORE DROP RENAME TO JOB_HISTORY_OLD;
```

# Recycle Bin leeren

## ■ Recycle Bin leeren

```
PURGE RECYCLEBIN;           -- Für aktuellen Benutzer
PURGE TABLE <table_name>;  -- Tabelle permanent löschen
PURGE INDEX <index_name>;   -- Index löschen
PURGE TABLESPACE <ts_name>; -- Recycle Bin in bestimmtem TS
                             löschen
PURGE TABLESPACE <ts_name> USER <username>;
                             -- Bestimmter User in Tablespace
PURGE DBA_RECYCLEBIN;       -- Leert gesamten Recycle Bin
```

## ■ Löschen ohne Recycle Bin

```
DROP TABLE TESTLADUNG PURGE;
```

# Einschränkungen Recycle Bin

---

- **Bei Platzmangel werden die Objekte dem Alter nach aus dem Recycle Bin gelöscht**
  - dann erst autoextend wenn aktiv
- **Größe und Vorhaltezeit nur durch freien Platz im Tablespace bestimmt**
  - keine Garantie möglich
- **Nicht möglich für System Tablespace**
- **Objekte im Recycle Bin nur lesbar**

# 5

## Flashback Database Die Rückspultaste für die DB

# Flashback Database (I)

---

- **Während des Normalbetriebs werden bei Änderungen Block-Images in die Flashback Logs kopiert**
- **Zeitbedarf für Flashback hängt nur vom Umfang der unerwünschten Änderungen ab**
- **Nur geänderte Blöcke werden aus der Flashback Area in umgekehrter Reihenfolge zurückkopiert**
- **Gesamtgröße der DB hat keinen Einfluss!**

# Flashback Database (II)

---

- **Das Ergebnis entspricht dem klassischen Point-In-Time-Recovery**
- **FLASHBACK DATABASE auch im RMAN möglich**
- **Ideal für Softwaretests, um immer wieder schnell auf einen definierten Zustand zurückzugehen**
- **Nutzung bei Standbydatenbanken nach Failover**

# Flashback Database aktivieren

---

## ■ Flash Recovery Area einrichten

```
DB_RECOVERY_FILE_DEST_SIZE  
DB_RECOVERY_FILE_DEST
```

## ■ DB muss im ARCHIVELOG Modus sein

## ■ DB\_FLASHBACK\_RETENTION\_TARGET Parameter bestimmt die Vorhaltezeit – Keine Garantie!

## ■ Im Mount Status:

```
ALTER DATABASE FLASHBACK ON;
```



# Flashback Database

---

```
FLASHBACK DATABASE TO  
(SCN | TIMESTAMP) | RESETLOGS | RESTORE POINT
```

## Beispiel

```
STARTUP MOUNT
```

```
FLASHBACK DATABASE TO  
TIMESTAMP to_timestamp  
( '2009-10-10 10:00:00' ,  
'YYYY-MM-DD HH24:MI:SS' )
```

```
ALTER DATABASE OPEN READ ONLY
```

➔ **Resultat überprüfen, ggf. weitere Flashback Kommandos**

```
ALTER DATABASE OPEN RESETLOGS
```

# Einschränkungen Flashback Database

---

- **Flashback Database ist nicht möglich, wenn:**
  - Ein Tablespace gelöscht worden ist
  - Eine Datendatei verkleinert worden ist
  - Die Controlfiles neu erstellt worden sind
  - Die Flashback Logs nicht mehr vorhanden sind

# 6

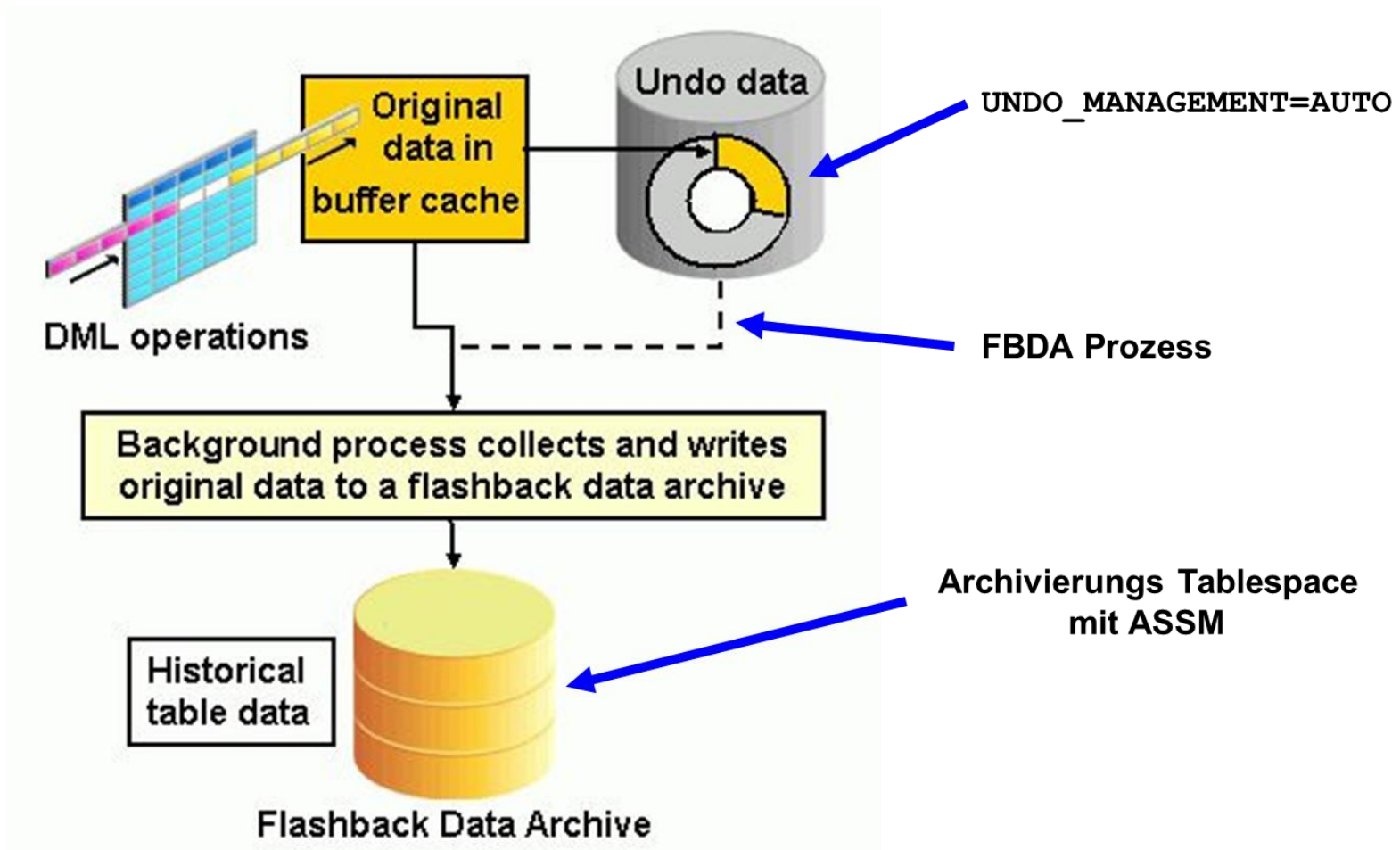
## Flashback Data Archive Total Recall

# Flashback Data Archive: Was ist das?

---

- **Kostenpflichtige Option der 11g Enterprise Edition**
- **Flashback Queries über UNDO\_RETENTION hinaus**
- **Langer Flashback Zeitraum mit wenig Platzbedarf, da auf Tabellenebene definiert**
- **Historisierte Daten werden als komprimierte Partitionen verwaltet**
- **Veränderungen werden auch über Jahre nachverfolgbar (betriebliche oder gesetzliche Anforderungen)**
- **Eigene Protokollierung über Trigger nicht mehr erforderlich**
- **Historische Daten werden nach der vorgegebenen Zeit automatisch entfernt**

# Architektur Flashback Data Archive



# Einrichtung Flashback Data Archive (I)

---

- **Einrichtung eines Tablespaces mit ASSM**

- **Erstellung eines FLASHBACK ARCHIVE Bereichs**

```
CREATE FLASHBACK ARCHIVE LONGARC  
TABLESPACE LANGZEIT  
RETENTION 5 YEAR; -- MONTH|DAY
```

- **Optional**

```
ALTER FLASHBACK ARCHIVE longarc DEFAULT;
```

- **Tabelle zu FLASHBACK ARCHIVE hinzufügen:**

```
ALTER TABLE meinetabelle FLASHBACK ARCHIVE longarc;
```

# Einschränkungen bei Flashback Data Archive

---

## ■ Bis inkl. 11.1:

- Kein DROP/TRUNCATE auf diese Tabelle
- Nur Spalten hinzufügen möglich

## ■ Ab 11.2 möglich:

- RENAME/DROP/TRUNCATE dieser Tabelle
- DROP/TRUNCATE Partition
- ADD/DROP/RENAME/MODIFY Column

## ■ Nicht möglich bei LONG Spalten, Nested Table Spalten, external/clustered/temporary Tabellen

# Fazit Flashback Allgemein

---

- **Leider ist vieles nur in der Enterprise Edition verfügbar**
- **Je nach benötigtem Zeitfenster erheblicher Platzbedarf für UNDO, Flash Recovery Area oder FBDA Tablespace**
- **Die Flashback „Familie“ bekommt mit jeder Version Zuwachs**
- **Interessante Erweiterung des DBA Werkzeugkastens**
- **Befehlssyntax relativ einfach**





# Ansprechpartner bei OPITZ CONSULTING

---

## **Daniel Schulz, Solution Architect**

OPITZ CONSULTING Gummersbach GmbH

[daniel.schulz@opitz-consulting.com](mailto:daniel.schulz@opitz-consulting.com)

Telefon +49 2261 6001 0

Mobil +49 173 7252413

## **Marion Mahr, Senior Consultant**

OPITZ CONSULTING München GmbH

[marion.mahr@opitz-consulting.de](mailto:marion.mahr@opitz-consulting.de)

Telefon +49 89 680098 0

Mobil +49 173 5549630