

**DOAG 2014**

---

**ORACLE LOGMINER**

**EIN NÜTZLICHES WERKZEUG**

**JOSEF LIPPERT  
FREIBERUFLICHER IT CONSULTANT  
MÜNCHEN**

---

# Wer bin ich ...

- Freiberuflicher IT Consultant
  - seit mehr als 14 Jahren
- Studium Mathematik
- IT Projekte im Bereich:
  - Oracle-DB (Beratung/Administration/Performance)
  - UNIX/Linux/Windows
  - RZ Infrastruktur Management
  - RZ Automation (z. B. Orsyp/Automic Dollar Universe)
  - Monitoring (z. B. BMC, HP, Microsoft)
- Email: [Josef.Lippert@jal-bs.de](mailto:Josef.Lippert@jal-bs.de)

# Agenda

---

- Einführung
- Integration OEM/Cloud Control
- Setup
- Einschränkungen
- Anwendungsbeispiele

# Einführung

- Alle DB-Änderungen → RedoLogs
- LogMiner liest die RedoLogs
- Einsatzmöglichkeiten:
  - Recovery
  - Analyse der Daten
  - Statistiken erstellen
  - Auditing

# Einführung

---

## Grundkonfiguration:

- Source Datenbank
- Mining Datenbank
- LogMiner Dictionary
- Menge aller RedoLogs (online, offline)

# Einführung

## Grundkonfiguration:

- Source und Mining Datenbank:
  - identische Hardware
  - können identisch sein
  - Release-Stand der Mining-DB  $\geq$  Source-DB
  - Character-Set muss identisch sein
  - LogMiner-Dictionary von der Source-DB

# Arbeitsschritte

- Aktivierung „SUPPLEMENTAL LOG DATA“
- Dictionary erstellen
- RedoLogs auswählen und registrieren
- Analyse durchführen
- Ergebnis: V\$LOGMNR\_CONTENTS

# Oracle Enterprise Manager

ORACLE Enterprise Manager 11g  
Database Control

Cluster Database:

Home Performance Availability Server Schema Data Movement Software and Su

High Availability Console

**Backup/Recovery**

**Setup**

- Backup Settings
- Recovery Settings
- Recovery Catalog Settings

**Manage**

- Schedule Backup
- Manage Current Backups
- Backup Reports
- Manage Restore Points
- Perform Recovery
- View and Manage Transactions

**Services**



# Oracle Cloud Control

The screenshot displays the Oracle Enterprise Manager Cloud Control 12c interface. At the top, the title bar reads "ORACLE Enterprise Manager Cloud Control 12c". Below this is a navigation bar with icons and labels for "Enterprise", "Ziele", "Favoriten", and "Historie". The main navigation area includes "Oracle-Datenbank", "Performance", "Verfügbarkeit", "Schema", and "Administration". The "Verfügbarkeit" menu is open, showing options: "High Availability-Konsole", "MAA-Advisor", "Backup und Recovery", and "Standby-Datenbank hinzufügen ...". The "Backup und Recovery" option is further expanded to show: "Backup planen ...", "Aktuelle Backups verwalten", "Backup-Berichte", "Restore Points", "Recovery ausführen ...", "Transaktionen", "Backup-Einstellungen", "Recovery-Einstellungen", and "Einstellungen für Recovery-Katalog".

**Zusammenfassung**

**Status**

Produktive Zeit	10 Tage, 3 Std.
Version	11.2.0.3.0
Laden	0,84 durchschnittliche aktive Sessions
Anzahl der Sessions	91
Letztes Backup	04.07.2014 12:01:36
Verfügbare Platz	259,96 GB
Gesamte SGA	5.097,13 MB

**Diagnose**

ADDM-Ergebnisse 6

# Supplemental Logging

- Database-Level Supplemental Logging
  - Minimal Supplemental Logging
    - ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
  - Identification Key Logging
    - Erforderlich, wenn RedoLogs nicht von der Source-DB
    - z.B. von der logical Standby-DB
    - ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;
- Table-Level Supplemental Logging
  - all, primary key, foreign key, unique key
    - ALTER TABLE HR.EMPLOYEES ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;

# Supplemental Logging

- SELECT supplemental\_log\_data\_min „STATUS“  
FROM V\$DATABASE;

```
STATUS  
-----  
YES
```

- SELECT supplemental\_log\_data\_min „Min“, supplemental\_log\_data\_pk „Primary“,  
supplemental\_log\_data\_ui „Unique“, supplemental\_log\_data\_fk „Foreign“,  
supplemental\_log\_data\_all „All“  
FROM V\$DATABASE;

```
Min      Primary  Unique   Foreign  All  
-----  ---      ---      ---      ---  
YES      NO       NO       NO       NO
```

# Supplemental Logging

---

## Hinweis

Das Aktivieren vom  
„Supplemental Logging“  
beeinflusst  
die Anzahl der erzeugten RedoLogs!

# Berechtigungen

- Keine sys/internal Berechtigung erforderlich
- GRANT EXECUTE ON SYS.DBMS\_LOGMNR TO <user>;
- GRANT EXECUTE\_CATALOG\_ROLE TO <user>;
- GRANT LOGMINING TO <user>;
- utl\_file\_dir: Berechtigung für den Export des Data-Dictionary

# Datenschutz

---

## Grundregel des deutschen Datenschutzes

„Die Verarbeitung personenbezogener Daten ist nur dann zulässig, soweit dies eine Rechtsvorschrift erlaubt oder der Betroffene eingewilligt hat (Bundesdatenschutz).“

# Table Storage Types

---

- Cluster Tabellen
- Index-organized Tabellen (IOTs)
- Heap-organized Tabellen
- Advanced compression (11.2)
- Tabellen mit LOBs (11.2)

# Unsupported Data Types

- Die komplette Tabelle wird ignoriert:
  - BFILE
  - Collections (including VARRAYs and nested tables)
  - Objects with nested tables and REFs
- SecureFiles LOBs: compatibility  $\geq$  11.2
- TDE, Compression: compatibility  $\geq$  11.1



# View: V\$LOGMNR\_CONTENTS

- Type: insert, update, delete, DDL
- SCN der Änderung
- SCN vom „commit“
- Transaktion-ID
- Tabelle, Schema
- Wer führte DDL, DML aus
- SQL, SQL\_UNDO
- Encrypted Passwort

# Setup

- alter database add supplemental log data;
- alter database add supplemental log data (primary key) columns;
  - **select supplemental\_log\_data\_min, supplemental\_log\_data\_pk  
from v\$database;**

SUPPLEMENTAL\_LOG\_DATA\_MIN

-----

YES

SUPPLEMENTAL\_LOG\_DATA\_PK

-----

YES

- **alter system switch logfile;**

# Analyse

- EXECUTE sys.DBMS\_LOGMNR.ADD\_LOGFILE(LOGFILENAME => '/u01/oradata/OR121C/fast\_recovery\_area/OR121C/archivelog/....arc', OPTIONS => DBMS\_LOGMNR.NEW);
- EXECUTE sys.DBMS\_LOGMNR.START\_LOGMNR(OPTIONS => DBMS\_LOGMNR.DICT\_FROM\_ONLINE\_CATALOG);
- SELECT username AS USR, (XIDUSN || '.' || XIDSLT || '.' || XIDSQN) AS XID, SQL\_REDO, SQL\_UNDO FROM V\$LOGMNR\_CONTENTS WHERE username IN ('MINER');
- EXECUTE DBMS\_LOGMNR.END\_LOGMNR();

# Statistiken/Auditing

- SELECT distinct username, SEG\_OWNER, seg\_name, operation  
from V\$LOGMNR\_CONTENTS  
where SEG\_OWNER not in ('SYS','SYSTEM')  
order by username, SEG\_OWNER, seg\_name, operation;

USERNAME	SEG_OWNER	SEG_NAME	OPERATION
MINER	MINER	DOAG_TABLE	DDL
MINER	MINER	DOAG_TABLE	DELETE
MINER	MINER	DOAG_TABLE	INSERT
MINER	MINER	DOAG_TABLE	UPDATE

# Statistiken/Auditing

- SELECT OPERATION, COUNT(\*) FROM V\$LOGMNR\_CONTENTS  
GROUP BY OPERATION;

OPERATION	COUNT(*)
-----	-----
DDL	1
START	13
COMMIT	10
ROLLBACK	2
UPDATE	6
DELETE	3
INTERNAL	9
INSERT	10

# Mining (globaler User)

sqlplus / as sysdba

SQL\*Plus: Release 12.1.0.1.0 Production on Mon Jul 21 12:54:40 2014

...

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
```

...

```
CREATE USER c##miner IDENTIFIED BY password1 CONTAINER=ALL;
```

```
grant connect, resource to c##miner;
```

```
grant EXECUTE_CATALOG_ROLE TO c##miner;
```

```
grant LOGMINING TO c##miner;
```

...

Benutzer wurde erstellt.

Benutzerzugriff (Grant) wurde erteilt.

Benutzerzugriff (Grant) wurde erteilt.

Benutzerzugriff (Grant) wurde erteilt.

# Beispiel 1

sqlplus ...

SQL\*Plus: Release 12.1.0.1.0 Production on Mon Jul 21 12:55:40 2014

...

```
create table doag_table (id number, name varchar2(256));
```

```
insert into doag_table select rownum,rownum||'miner',sysdate from all_objects
```

```
where rownum<=2;
```

```
commit;
```

```
delete from doag_table;
```

```
rollback;
```

```
update doag_table set name='changed_name';
```

# Analyse aus V\$LOGMNR\_CONTENTS

In den folgenden Beispielen wurde das Ergebnis mit folgendem Statement erstellt:

```
SELECT    username AS USR,  
          (XIDUSN || '.' || XIDSLT || '.' || XIDSQN) AS XID,  
          SQL_REDO,  
          SQL_UNDO  
FROM V$LOGMNR_CONTENTS  
WHERE username IN ('MINER');
```



# Beispiel 1 (cont.)

--> SQL: create table doag\_table (id number, name varchar2(256));

Ergebnis aus V\$LOGMNR\_CONTENTS:

...

MINER 14.23.11

create table doag\_table (id number, name varchar2(256));

“no undo statement”

...

MINER 14.23.11

update "SYS"."OBJ\$" set "OBJ#" = '91620', "DATAOBJ#" = '91620', "TYPE#" = '2',...

....

“no undo statement”

# Beispiel 1 (cont.)

--> SQL: insert into doag\_table select rownum,rownum||'miner' from all\_objects where rownum<=2;

Ergebnis aus V\$LOGMNR\_CONTENTS:

MINER 5.0.1773

insert into "MINER"."DOAG\_TABLE"("ID","NAME") values ('1','1miner');

delete from "MINER"."DOAG\_TABLE" where "ID" = '1' and "NAME" = '1miner' and ROWID = 'AAAWXkAAKAAAACDAAA';

MINER 5.0.1773

insert into "MINER"."DOAG\_TABLE"("ID","NAME") values ('2','2miner');

delete from "MINER"."DOAG\_TABLE" where "ID" = '2' and "NAME" = '2miner' and ROWID = 'AAAWXkAAKAAAACDAAB';

# Beispiel 1 (cont.)

---

--> SQL: commit;

Ergebnis aus V\$LOGMNR\_CONTENTS:

MINER                    5.0.1773

commit;

“no undo statement”

# Beispiel 1 (cont.)

--> SQL: delete from doag\_table;

Ergebnis aus V\$LOGMNR\_CONTENTS:

MINER 2.23.1806

set transaction read write;

MINER 2.23.1806

delete from "MINER"."DOAG\_TABLE" where "ID" = '1' and "NAME" = '1miner' and ROWID = 'AAAWXkAAKAAAACDAAA';

insert into "MINER"."DOAG\_TABLE"("ID","NAME") values ('1','1miner');

MINER 2.23.1806

delete from "MINER"."DOAG\_TABLE" where "ID" = '2' and "NAME" = '2miner' and ROWID = 'AAAWXkAAKAAAACDAAB';

insert into "MINER"."DOAG\_TABLE"("ID","NAME") values ('2','2miner');

# Beispiel 1 (cont.)

---

--> SQL: rollback;

Ergebnis aus V\$LOGMNR\_CONTENTS:

MINER                    2.23.1806

rollback;

“no undo statement”

# Beispiel 1 (cont.)

--> SQL: update doag\_table set name='changed\_name';

Ergebnis aus V\$LOGMNR\_CONTENTS:

```
MINER                1.22.1420
update "MINER"."DOAG_TABLE" set "NAME" = 'changed_name' where "NAME" = '1miner'
and ROWID = 'AAAWXkAAKAAAACDAAA';
update "MINER"."DOAG_TABLE" set "NAME" = '1miner' where "NAME" = 'changed_name'
and ROWID = 'AAAWXkAAKAAAACDAAA';
```

```
MINER                1.22.1420
update "MINER"."DOAG_TABLE" set "NAME" = 'changed_name' where "NAME" = '2miner'
and ROWID = 'AAAWXkAAKAAAACDAAB';
update "MINER"."DOAG_TABLE" set "NAME" = '2miner' where "NAME" = 'changed_name'
and ROWID = 'AAAWXkAAKAAAACDAAB';
```

# Beispiel 2

sqlplus ...

SQL\*Plus: Release 12.1.0.1.0 Production on Mon Jul 21 12:56:40 2014

...

```
CREATE TABLE DOAG_TABLE2 (c1 NUMBER, c2 NUMBER, c3 NUMBER);
```

```
ALTER TABLE DOAG_TABLE2 ADD CONSTRAINT DOAG_TABLE2_PK PRIMARY KEY  
(c1,c2);
```

```
INSERT INTO DOAG_TABLE2 VALUES (1,2,3);
```

```
UPDATE DOAG_TABLE2 SET c3 = 30 WHERE c1 = 1 AND c2 = 2;
```

# Beispiel 2 (cont.)

--> SQL: CREATE TABLE DOAG\_TABLE2 (c1 NUMBER, c2 NUMBER, c3 NUMBER);

Ergebnis aus V\$LOGMNR\_CONTENTS:

...

MINER 2.30.1816

CREATE TABLE DOAG\_TABLE2 (c1 NUMBER, c2 NUMBER, c3 NUMBER);

“no undo statement”

MINER 2.30.1816

update "SYS"."OBJ\$" set "OBJ#" = '91630', "DATAOBJ#" = '91630', "TYPE#" = '2',...

....

“no undo statement”

...

commit;



# Beispiel 2 (cont.)

→SQL: ALTER TABLE DOAG\_TABLE2 ADD CONSTRAINT DOAG\_TABLE2\_PK  
PRIMARY KEY (c1,c2);

Ergebnis aus V\$LOGMNR\_CONTENTS:

```
MINER                14.18.23
CREATE UNIQUE INDEX "MINER"."DOAG_TABLE2_PK" on "MINER"."DOAG_TABLE2"("C1","C2")
NOPARALLEL;
"no undo statement"
...
MINER                14.18.23
insert into "SYS"."ICOL$"("OBJ#...
"no undo statement"
...
MINER                14.18.23
ALTER TABLE DOAG_TABLE2 ADD CONSTRAINT DOAG_TABLE2_PK PRIMARY KEY (c1,c2);
"no undo statement"
...
commit;
```

# Beispiel 2 (cont.)

→SQL: INSERT INTO DOAG\_TABLE2 VALUES (1,2,3);

Ergebnis aus V\$LOGMNR\_CONTENTS:

MINER                    12.3.29

insert into "MINER"."DOAG\_TABLE2"("C1","C2","C3") values ('1','2','3');

delete from "MINER"."DOAG\_TABLE2" where "C1" = '1' and "C2" = '2' and "C3" = '3' and ROWID = 'AAAWXuAAKAAAACLAAA';

# Beispiel 2 (cont.)

→SQL: UPDATE DOAG\_TABLE2 SET c3 = 30 WHERE c1 = 1 AND c2 = 2;

Ergebnis aus V\$LOGMNR\_CONTENTS:

MINER 12.3.29

```
update "MINER"."DOAG_TABLE2" set "C3" = '30' where "C1" = '1' and "C2" = '2' and "C3" = '3'
and ROWID = 'AAAWXuAAKAAAACLAAA';
```

```
update "MINER"."DOAG_TABLE2" set "C3" = '3' where "C1" = '1' and "C2" = '2' and "C3" = '30'
and ROWID = 'AAAWXuAAKAAAACLAAA';
```

# Beispiel 3

**Welche Änderungen wurden innerhalb der letzten Stunde durchgeführt:**

->SQL: SELECT timestamp, scn, sql\_redo, sql\_undo,row\_id, seg\_name,  
seg\_owner, username from v\$logmnr\_contents  
where seg\_owner='MINER'  
and timestamp between sysdate-1/24 and sysdate;

24.07.2014-14:31:03 2050890

create table doag\_table (id number, name varchar2(256));

AAAAAAAAAAAAAAAAAAB DOAG\_TABLE MINER MINER

24.07.2014-14:31:06 2050902

insert into "MINER"."DOAG\_TABLE"("ID","NAME") values ('1','1miner');

delete from "MINER"."DOAG\_TABLE" where "ID" = '1' and "NAME" = '1miner' and ROWID = 'AAAWXkAAKAAAACDAAA';

AAAWXkAAKAAAACDAAA DOAG\_TABLE MINER MINER

24.07.2014-14:31:06 2050902

insert into "MINER"."DOAG\_TABLE"("ID","NAME") values ('2','2miner');

delete from "MINER"."DOAG\_TABLE" where "ID" = '2' and "NAME" = '2miner' and ROWID = 'AAAWXkAAKAAAACDAAB';

AAAWXkAAKAAAACDAAB DOAG\_TABLE MINER MINER

# Beispiel 4

## In welchem Zeitintervall erfolgt die Analyse:

->SQL: select min(timestamp), max(timestamp) from V\$logmnr\_contents;

MIN(TIMESTAMP)

16-APR-2014 02:34:15

MAX(TIMESTAMP)

16-APR-2014 11:00:42

## Welche Objekte wurden modifiziert:

->SQL: SELECT distinct seg\_name from v\$logmnr\_contents;

SEG\_NAME

...

"MINER"."DOAG\_TABLE2"

...

# Zusammenfassung

- Oracle LogMiner ist ein kostenfreies, nützliches Werkzeug
- Analyse der RedoLogs (online, offline)
- Einfache Bedienung
- Einsatz als Recovery-Tool
- Erstellen von Statistiken
- Auditing

# Oracle LogMiner - ein nützliches Werkzeug

---

Danke für Ihre Aufmerksamkeit

Fragen ...