

Kuriositäten in der Oracle-Datenbank

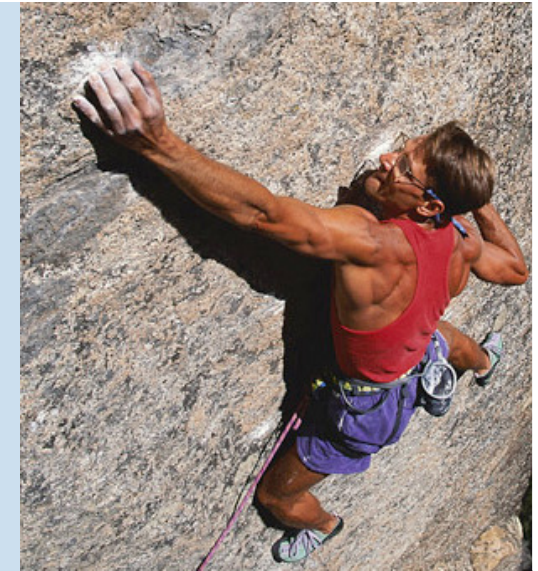
19. Deutsche ORACLE-Anwenderkonferenz

Do. 16.11., 14.00 Uhr, Variohalle 1

Dr. Peter Alteheld, Systemberater

MT AG, Bereich Solutions Development, FB Plattform Services

Ratingen bei Düsseldorf



Unerwartetes Verhalten von Software

- **Überraschung**
- **Fehlerhafte Ergebnisse oder gar Inkonsistenzen**
- **Aufwand**

Ursachenbekämpfung?

- **Das Verhalten kennen → Aneignung von Wissen**
- **Software an Erwartungen anpassen?**

NULL-Werte

- Abweichende Summen
- Abweichende Vergleichsergebnisse

Constraints

- Readonly-Modus per Constraint
- Löschung ohne Auswirkung

PL/SQL

- Andere Berechtigungen als in SQL
- Doppelt zündender Trigger
- Deadlock innerhalb einer Sitzung

NULL-Werte

- **Abweichende Summen**
- **Abweichende Vergleichsergebnisse**

Constraints

- **Readonly-Modus per Constraint**
- **Löschung ohne Auswirkung**

PL/SQL

- **Andere Berechtigungen als in SQL**
- **Doppelt zündender Trigger**
- **Deadlock innerhalb einer Sitzung**

Vergleich von Addition und Summenfunktion

```
SQL> create table numbers (x NUMBER, y NUMBER, z NUMBER);
```

```
SQL> insert into numbers values (1,4,NULL);
```

```
SQL> insert into numbers values (2,NULL,NULL);
```

```
SQL> select y+z from numbers;
```

Y+Z

(NULL)

(NULL)

```
SQL> select sum(y), sum(z) from numbers;
```

SUM(Y)

4

SUM(Z)

(NULL)

Selbstdefinierte Summenfunktion

- User-Defined Aggregate Function – Kap. 12 im Oracle Data Cartridge Developer's Guide

Implementation der selbstdefinierten Summenfunktion mysum():

```
CREATE OR REPLACE TYPE mysumImpl AS OBJECT (  
  summe NUMBER,  
  STATIC FUNCTION odciaggregateinitialize(sctx IN OUT mysumImpl)  
    RETURN NUMBER,  
  MEMBER FUNCTION odciaggregateiterate(SELF IN OUT  
    mysumImpl, VALUE IN NUMBER) RETURN NUMBER,  
  MEMBER FUNCTION odciaggregateterminate(SELF IN OUT mysumImpl,  
    returnvalue OUT NUMBER, flags IN NUMBER) RETURN NUMBER,  
  MEMBER FUNCTION odciaggregatemerge(SELF IN OUT mysumImpl, ctx2  
    IN mysumImpl) RETURN NUMBER);
```

```
CREATE OR REPLACE TYPE BODY mysumImpl
IS
  STATIC FUNCTION odciaggregateinitialize(sctx IN OUT mysumImpl)
  RETURN NUMBER
  IS BEGIN sctx := mysumImpl(0); RETURN odciconst.success;
  END;
  MEMBER FUNCTION odciaggregateiterate(SELF IN OUT
  mysumImpl, VALUE IN NUMBER) RETURN NUMBER
  IS BEGIN SELF.summe := SELF.summe + NVL(VALUE, 0);
  RETURN odciconst.success; END;
  MEMBER FUNCTION odciaggregateterminate(SELF IN OUT mysumImpl,
  returnvalue OUT NUMBER, flags IN NUMBER) RETURN NUMBER
  IS BEGIN returnvalue := SELF.summe; RETURN
  odciconst.success; END;
  MEMBER FUNCTION odciaggregatemerge(SELF IN OUT mysumImpl, ctx2
  IN mysumImpl) RETURN NUMBER
  IS BEGIN RETURN odciconst.success; END;
END;
```

Selbstdefinierte Summenfunktion mysum()

```
CREATE OR REPLACE FUNCTION mysum (input NUMBER) RETURN NUMBER  
AGGREGATE USING mysumImpl;
```

```
SQL> select mysum(y), mysum(z) from numbers;
```

MYSUM (Y)	MYSUM (Z)
----- 4	----- 0

NULL-Werte

- Abweichende Summen
- **Abweichende Vergleichsergebnisse**

Constraints

- Readonly-Modus per Constraint
- **Löschung ohne Auswirkung**

PL/SQL

- Andere Berechtigungen als in SQL
- **Doppelt zündender Trigger**
- **Deadlock innerhalb einer Sitzung**

NULL ungleich NULL

```
SQL> declare
  2  a NUMBER;
  3  b NUMBER;
  4  begin
  5  a := NULL;
  6  b := NULL;
  7  if (a=b)
  8  then dbms_output.put_line('NULL=NULL');
  9  else dbms_output.put_line('NULL<>NULL');
 10  end if;
 11  end;
 12  /
NULL<>NULL
```

NULL gleich NULL

```
SQL> create table unterkonten(kontonr number(7), uktonr
    number(2), ukto_text varchar2(30));
SQL> insert into unterkonten values (1234567, NULL, 'Für
    Dividendenauszahlungen');
SQL> commit;
SQL> create unique index ukto_idx on unterkonten(kontonr,
    uktonr);

SQL> insert into unterkonten values (1234567, NULL, 'Für Tilgung
    Hauskauf-Kredit');
insert into unterkonten values (1234567, NULL, 'Für Tilgung
    Hauskauf-Kredit')
*
FEHLER in Zeile 1:
ORA-00001: Unique Constraint (TESTUSER.UKTO_IDX) verletzt
```

NULL-Werte

- Abweichende Summen
- Abweichende Vergleichsergebnisse

Constraints

- **Readonly-Modus per Constraint**
- Löschung ohne Auswirkung

PL/SQL

- Andere Berechtigungen als in SQL
- Doppelt zündender Trigger
- Deadlock innerhalb einer Sitzung

```
SQL> CREATE TABLE test (id NUMBER);
```

```
SQL> ALTER TABLE test ADD CONSTRAINT test_read_only CHECK(1=1)  
      DISABLE VALIDATE;
```

```
SQL> INSERT INTO test VALUES (1);
```

```
INSERT INTO test VALUES (1)
```

```
*
```

```
FEHLER in Zeile 1:
```

```
ORA-25128: Kein Einfügen/Aktualisieren/Löschen bei Tabelle mit  
deaktiviertem
```

```
und validiertem Constraint (TESTUSER.TEST_READ_ONLY)
```

Vorteile gegenüber Tabelle in Readonly-Tablespace

- schnell, auch bei großen Tabellen
- auch schnell zurückzunehmen
- Anwendungen, die Tablespaces zuweisen

Nachteile

- manch erfahrener ORACLE-Nutzer kennt dieses Feature nicht

NULL-Werte

- Abweichende Summen
- Abweichende Vergleichsergebnisse

Constraints

- Readonly-Modus per Constraint
- **Löschung ohne Auswirkung**

PL/SQL

- Andere Berechtigungen als in SQL
- Doppelt zündender Trigger
- Deadlock innerhalb einer Sitzung

Constraints: Löschung ohne Auswirkung



```
SQL> ALTER TABLE test ADD CONSTRAINT willi CHECK(id>4) ENABLE  
VALIDATE;
```

```
SQL> ALTER TABLE test ADD CONSTRAINT willi2 CHECK(id>4) ENABLE  
VALIDATE;
```

```
SQL> INSERT INTO test VALUES (1);  
INSERT INTO test VALUES (1)
```

*

FEHLER in Zeile 1:

ORA-02290: CHECK-Constraint (TESTUSER.WILLI2) verletzt

```
SQL> ALTER TABLE test DROP CONSTRAINT willi2;
```

```
SQL> INSERT INTO test VALUES (1);  
INSERT INTO test VALUES (1)
```

*

FEHLER in Zeile 1:

ORA-02290: CHECK-Constraint (TESTUSER.WILLI) verletzt

Constraints: Löschung ohne Auswirkung



```
CREATE FUNCTION search_cond(p_cons_name IN VARCHAR2) RETURN VARCHAR2
AS
  l_search_condition VARCHAR2(4000);
BEGIN
  SELECT search_condition INTO l_search_condition
  FROM user_constraints WHERE constraint_name = p_cons_name;
  RETURN l_search_condition;
END;
```

```
SQL> SELECT search_cond_vc, COUNT(*)
  2 FROM (SELECT search_cond(constraint_name) search_cond_vc
  3        FROM user_constraints
  4        WHERE constraint_name like '%')
  5 GROUP BY search_cond_vc
  6 HAVING COUNT(*) > 1;
```

SEARCH_COND_VC	COUNT(*)
-----	-----
id>4	2

NULL-Werte

- Abweichende Summen
- Abweichende Vergleichsergebnisse

Constraints

- Readonly-Modus per Constraint
- Löschung ohne Auswirkung

PL/SQL

- **Andere Berechtigungen als in SQL**
- Doppelt zündender Trigger
- Deadlock innerhalb einer Sitzung

Berechtigungen in SQL

```
SQL> create user testuser identified by testuser;
```

```
SQL> grant dba to testuser;
```

```
SQL> conn testuser
```

```
SQL> create table system.test (x number);
```

```
SQL> insert into system.test values (4);
```

```
SQL> commit;
```

```
SQL> delete from system.test;
```

```
1 Zeile wurde gelöscht.
```

Berechtigungen in anonymen PL/SQL-Block

```
SQL> rollback;
```

Transaktion mit ROLLBACK rückgängig gemacht.

```
SQL> begin
```

```
  2  delete from system.test;
```

```
  3  commit;
```

```
  4  end;
```

```
  5  /
```

PL/SQL-Prozedur wurde erfolgreich abgeschlossen.

```
SQL> select * from system.test;
```

Es wurden keine Zeilen ausgewählt

Berechtigungen in PL/SQL-Prozedur

```
SQL> create procedure del_system_test
  2  as
  3  begin
  4  delete from system.test;
  5  commit;
  6  end;
  7  /
```

Warnung: Prozedur wurde mit Kompilierungsfehlern erstellt.

```
SQL> sho err
```

```
Fehler bei PROCEDURE DEL_SYSTEM_TEST:
```

```
LINE/COL ERROR
```

```
-----
4/1      PL/SQL: SQL Statement ignored
4/20     PL/SQL: ORA-00942: Tabelle oder View nicht vorhanden
```

Oracle Database Application Developer's Guide – Fundamentals

*Note: To create without errors (**to compile the procedure** or **package successfully**) requires the following additional privileges:*

- *The owner of the procedure or package must be explicitly granted the necessary object privileges for all objects referenced within the body of the code.*
- ***The owner cannot obtain required privileges through roles.***

Kompilierter Code wäre nicht mehr valid, wenn Rollen entzogen würden.

SQL-Test

```
SQL> set role none;
```

Rolle wurde gesetzt.

```
SQL> delete from system.test;
```

```
delete from system.test
```

*

FEHLER in Zeile 1:

```
ORA-00942: Tabelle oder View nicht vorhanden
```

Erforderliche Berechtigungen für das Ausführen einer Prozedur

- Bei Erstellung mit den Rechten des Prozedureigentümers (Default, AUTHID DEFINER):
 - Das Objektprivileg EXECUTE für die Prozedur

- Bei Erstellung mit den Rechten des Ausführenden (AUTHID CURRENT_USER):
 - Das Objektprivileg EXECUTE für die Prozedur und
 - explizite Grants auf alle in der Prozedur angesprochenen Objekte
 - ODER diese Berechtigungen über Rollenzuweisung

NULL-Werte

- Abweichende Summen
- Abweichende Vergleichsergebnisse

Constraints

- Readonly-Modus per Constraint
- Löschung ohne Auswirkung

PL/SQL

- Andere Berechtigungen als in SQL
- **Doppelt zündender Trigger**
- Deadlock innerhalb einer Sitzung

```
CREATE TABLE t (x INT, y INT);
INSERT INTO t VALUES (1,1);
COMMIT;
CREATE OR REPLACE TRIGGER t_bufer
BEFORE UPDATE ON t FOR EACH ROW
BEGIN
  DBMS_OUTPUT.PUT_LINE('old.x=, || :old.x || ', old.y=' ||
    :old.y);
  DBMS_OUTPUT.PUT_LINE('new.x=' || :new.x || ', new.y=' ||
    :new.y);
END;
```

Sitzung 1:

```
SQL> UPDATE t SET x=x+1;
old.x=1, old.y=1 -- t_bufer zündet 1x
new.x=2, new.y=1
1 Zeile wurde aktualisiert.
```

Sitzung 2:

```
SQL> UPDATE t SET x=x+1 WHERE x>0; -- HÄNGT
```

Sitzung 1:

```
SQL> commit;  
Transaktion mit COMMIT abgeschlossen.
```

Sitzung 2:

```
old.x=1, old.y=1      -- t_bufer zündet zum 1.Mal  
new.x=2, new.y=1  
old.x=2, old.y=1      -- t_bufer zündet zum 2.Mal  
new.x=3, new.y=1
```

1 Zeile wurde aktualisiert.

- In Sitzung 2 erfolgt nach einem Rollback ein Restart (Schreib-Konsistenz)
 - Im Trigger nur Code verwenden, der zurückgerollt werden kann, der also transaktional ist. Nicht-transaktional sind z.B. DBMS_OUTPUT, die UTL-Packages, autonome Transaktionen, das Ändern von Packagevariablen.

NULL-Werte

- Abweichende Summen
- Abweichende Vergleichsergebnisse

Constraints

- Readonly-Modus per Constraint
- Löschung ohne Auswirkung

PL/SQL

- Andere Berechtigungen als in SQL
- Doppelt zündender Trigger
- **Deadlock innerhalb einer Sitzung**

Ausschnitt aus dem Trace-File:

DEADLOCK DETECTED

Current SQL statement for this session:

```
INSERT INTO RATE_PROTOKOLL ( ID,KD_STAMMNR,..., :b9 )
```

----- PL/SQL Call Stack -----

...

The following deadlock is not an ORACLE error...

Deadlock graph:

Resource Name	process	session	holds	waits	process	session	holds	waits
TX-00050...	11	13	X		11	13		S

session 13: DID 0001-000B-0000A7F4

session 13: DID 0001-000B-0000A7F4

Rows waited on:

Session 13: no row

INSERTs in Master- und Detailtabelle in einer Sitzung:

```
SQL> insert into dept values (50, 'HEADQUARTER', 'WASHINGTON');  
1 Zeile wurde erstellt.
```

```
SQL> insert into emp values(6543, 'MORGAN', NULL, NULL, NULL,  
NULL, NULL, 50);  
1 Zeile wurde erstellt.
```

```
SQL> commit;  
Transaktion mit COMMIT abgeschlossen.
```

INSERTs in Master- und Detailtabelle in zwei Sitzungen:

Sitzung 1:

```
SQL> insert into dept values (60, 'EMEA HQ', 'BERLIN');
```

1 Zeile wurde erstellt.

Sitzung 2:

```
SQL> insert into emp values (8136, 'SCHULZE', NULL, NULL, NULL,  
NULL, NULL, 60); -- HÄNGT
```

INSERT in Detailtabelle in autonomer Transaktion:

```
CREATE OR REPLACE PROCEDURE pr_insert_emp(deptno number)
IS PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN insert into emp values(4711, 'MAIER', NULL, NULL, NULL, NULL,
NULL, deptno);
END;

CREATE OR REPLACE TRIGGER dept_as_iu
AFTER INSERT ON dept FOR EACH ROW
BEGIN pr_insert_emp(:new.deptno); END;

SQL> insert into dept values (70, 'TEST', 'MIAMI');
insert into dept values (70, 'TEST', 'MIAMI')
      *
```

FEHLER in Zeile 1:
ORA-00060: **Deadlock** beim Warten auf Ressource festgestellt
ORA-06512: in "SCOTT.PR_INSERT_EMP", Zeile 4
ORA-06512: in "SCOTT.DEPT_AS_IU", Zeile 2
ORA-04088: Fehler bei der Ausführung von Trigger 'SCOTT.DEPT_AS_IU'

- Summierungsfunktion weicht von Addition ab
- Aggregatfunktionen können selbst geschrieben werden
- NULL ist nicht immer ungleich NULL
- Disabled validated constraint setzt Tabelle in Lesemodus
- Rollen können zum Ausführen einer Prozedur reichen, aber nicht zum Kompilieren
- Before-Update-Trigger können mehrfach zünden
- In Triggern nur transaktionale Aktionen verwenden
- Deadlocks können innerhalb einer Sitzung auftreten

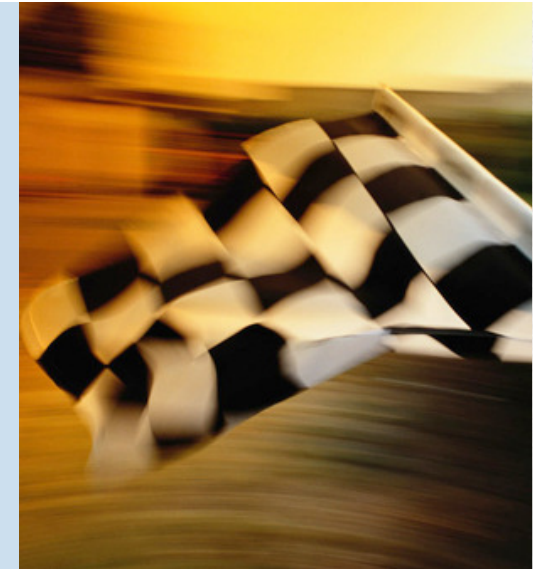
Fragen und Antworten

Dr. Peter Alteheld

MT AG · Balcke-Dürr-Allee 9 · 40882 Ratingen

Tel. 02102 309 61-0 · Fax 02102 309 61-10

www.mt-ag.com · dr.peter.alteheld@mt-ag.com



- Björn Bröhl:
„Best Practices für Installation und Betrieb des Oracle Application Servers unter Linux“
- Niels de Bruijn:
„Die WM 2006 beim WDR - Zusammenarbeit mit Portal und Web Conferencing“
- Sascha Hartmann:
„Ablösung einer Microsoft Umgebung durch die Oracle Collaboration Suite“
- Volker Mach:
„ORA-600 und ORA-7445 in der Oracle Datenbank“