

So was geht auch mit SQL?

Bekanntes und Unbekanntes aus dem SQL-Reich



Bruno Cirone

Themenverantwortlicher Security

Wer ist Bruno Cirone?

- Oracle Erfahrung seit 1986
- Selbständig seit 1989
- Projekte im In- und Ausland.
- Oracle, Adabas-D (MAXDB), Unix, Linux
- Mitglied im Planungs-/Entwicklungsteam von Adabas-D
- Migrationen von verschiedenen DB-Herstellern nach Oracle
- Sehr viel Spaß an Oracle Herausforderungen
- www.cirone.de
- [Email: Bruno@Cirone.de](mailto: Bruno@Cirone.de)

Motivation

Mein Erstkontakt mit dem Kunden:

„Wir haben ein Softwarehaus beauftragt bei der Umstellung unserer Daten (u.a. Adressdaten) alle Großbuchstaben in Klein- und Großbuchstaben umzuwandeln. Z.B. aus „HANS MAIER“ soll „Hans Meier“ werden. Die Erstellung der Programme hat bisher ca. **3 Wochen** gedauert und hat uns bisher ca. **15.000 €** gekostet. Die ersten Tests auf unseren Datenbestand laufen ca. **24 Stunden**. Können Sie die Laufzeit verbessern?“

Motivation

Meine Antwort:

Ja durch ein einfaches SQL Statement wie z.B.

```
update Adresse  
set Name      =Initcap(Name) ,  
    Vorname=Initcap(Vorname) usw.
```

Dieses Statement dauerte ca. 10 Minuten Entwicklungszeit.
Laufzeit auf den Datenbestand ca. 10 Minuten.

Kosten ca. 2



Rahmenbedingungen

- Die Beispiele und SQL-Statements sind vereinfacht.
- Das Prinzip sollte immer erkennbar sein.
- Minimalinvasiv
 - Nichts anfassen
 - Nichts ändern
 - Nichts anschauen
 - Usw.
- Nur Standardfunktionen und/oder Optionen nutzen
 - Keine Lizenzpflichtigen Optionen nutzen
 - Keine Zusatzsoftware
 - Usw.
- Also am besten mit „Hand Auflegen“ die Probleme lösen!

1. Marketingmäßige Sortierungen

Kunde:

- Beispiel von einem Kaufhaus
- In ähnlicher Form auch bei vielen anderen Kunden.

Problem:

- Es muss eine besondere Sortierreihenfolge implementiert werden, die nicht einfach aufsteigend oder absteigend ist. In diesem Fall müssen die Verkäufe nach Etagen, Warengruppen und Abteilung sortiert werden.

1. Marketingmäßige Sortierungen

Aus der Tabelle:

DATUM	ABTEILUNG	UMSATZ
22-APR-16	100	150
22-APR-16	101	250
22-APR-16	102	450
22-APR-16	103	650
22-APR-16	200	150
22-APR-16	201	250

Soll werden:

```
select *  
from Umsatz_view;
```

DATUM	ABTEILUNG	ETAGE	WARENGRUP	UMSATZ
22-APR-16	100	1	Herren	150
22-APR-16	101	2	Herren	250
22-APR-16	102	2	Herren	450
22-APR-16	103	9	Sonstiges	650

1. Marketingmäßige Sortierungen

Demo

2. Sanftes Löschen

Kunde:

- Beispiel von einem Papierhersteller

Problem:

- Plattenplatz und Datenbank sind nahezu zu 100% voll. Kein weiterer Plattenplatz kann angefügt werden.

2. Sanftes Löschen

`df -h`

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda2	400G	398.0G	2.0G	100%	/ora1
/dev/sda3	400G	396.0G	4.0G	99%	/ora2
/dev/sda4	400G	392.0G	8.0G	98%	/ora3
/dev/sda5	10G	5.0G	5.0G	50%	/ora_redo
/dev/sda6	100G	80.0G	20.0G	80%	/ora_archive

2. Sanftes Löschen

- Welches sind die größten Segmente?

```
select owner, segment_name, segment_type, bytes/1024/1024/1024  gb
       from dba_segments
       where bytes > (Select max(bytes) * 0.9
                      from dba_segments);
```

OWNER	SEGMENT_NAME	SEGMENT_TYPE	GB
-----	-----	-----	-----
SCHNITT	SCHNITT_WERTE	TABLE	428.28

- Die dazugehörigen Indices

OWNER	SEGMENT_NAME	SEGMENT_TYPE	GB
-----	-----	-----	-----
SCHNITT	SCHNITT_WERTE_I1	INDEX	78.76
SCHNITT	SCHNITT_WERTE_I2	INDEX	126.88

2. Sanftes Löschen

- Alle Daten, die älter sind als 90 Tage können gelöscht werden
- Kein Cronjob möglich (keine Loginshell)
- Keine Jobs in der Datenbank
- Maximal 5% pro Tag löschen
- Indices sollen auch kleiner werden
- Standard Edition, daher kein Online Rebuild möglich
- Keine Downtime

2. Sanftes Löschen

- Lösung mit rman

Sicherung der Datenbank

```
SQL "DELETE SCHNITT_WERTE WHERE erstellt_timestamp < sysdate - 90  
      AND ROWNUM <= (SELECT count(*) FROM SCHNITT_WERTE) * 0.05";
```

```
SQL "ALTER TABLE SCHNITT_WERTE SHRINK SPACE CASCADE";
```

3. Mehrfaches Replace

Kunde:

- Beispiel von einem Kreditversicherer

Problem:

- Der Name einer Firma muss von Zusätzen befreit werden. Z.B. aus „Gebr. Maier KG“ soll nur „Maier“ ausgegeben werden.

3. Mehrfaches Replace

Aus der Spalte Name:

NAME

Gebr. Maier
Schulze GmbH
Willi & Co. KG
Gebr. Mertens & Co. GmbH KGaA
Maier KG
Ludwig OHG

Soll werden:

NAME

Maier
Schulze
Willi
Mertens
Maier
Ludwig

3. Mehrfaches Replace

Bisheriges SQL:

```
SELECT REPLACE ( REPLACE ( REPLACE ( REPLACE ( REPLACE ( REPLACE (
REPLACE (REPLACE (name, 'Gebr. ', ''), ' Co.', ''), ' KGaA', ''), ' KG',
''), ' GmbH', ''), '&', ''), ' OHG', ' '), ' ', ' ') Name
FROM firma
```

3. Mehrfaches Replace

Oder Formatiert etwas lesbarer:

```
SELECT REPLACE (
    REPLACE (
        REPLACE (
            REPLACE (
                REPLACE (
                    REPLACE (
                        REPLACE (REPLACE (name, 'Gebr. ', ''), ' Co.', ''),
                        ' KGaA',
                        ''),
                    ' KG',
                    ''),
                ' GmbH',
                ''),
            '&',
            ''),
        ' OHG',
        ''),
    ' ',
    ' ')
Name
FROM firma
```

3. Mehrfaches Replace

- Lösung mit Regexpr_Replace

```
SELECT REGEXP_REPLACE (NAME, 'Gebr.|Co.|KGaA|KG|GmbH|&|OHG| ', '') NAME  
FROM FIRMA;
```

NAME

Maier
Schulze
Willi
Mertens
Maier
Ludwig

4. Datengenerierung

Kunde:

- Beispiel : Ticketing System (Berechnungen von SLA)

Problem:

- Es muss ein Tabelle aufgebaut werden, die alle Sekunden des aktuellen und des nächsten Jahres beinhaltet.

4. Datengenerierung

- Das Java Programmfragment

```
Statement batchStatement = connection.createStatement();

Calendar to = Calendar.getInstance();
to.setTime(converttoUTC(toLocal.getTime()));

Calendar from = Calendar.getInstance();
from.setTime(converttoUTC(fromLocal.getTime()));

String startStatement = "insert into "+tableName+" (timesec) values
(to date('";
String endStatement = "','"+yyyy-mm-dd hh24:mi:ss+"''))";

while(to.getTimeInMillis() >= from.getTimeInMillis()) {
    String query =
startStatement+sdf.format(from.getTime())+endStatement;
    batchStatement.addBatch(query);
}
batchStatement.executeBatch();
connection.commit();
```

4. Datengenerierung

- Besser geht es mit folgenden Statement:

```
create table x1 as
  select to_date('31.12.2015 23:59:59')+rownum / 86400 Sekunden
         from dual
connect by level <= 86400 * ( to_date('31.12.2017 23:59:59', 'dd.mm.yyyy hh24:mi:ss')
                             - to_date('31.12.2015 23:59:59', 'dd.mm.yyyy hh24:mi:ss')
                             ) /* Das entspricht 731 Tagen */
;
```

Elapsed: 00:03:20.60

5. Datumsproblematik (Motivation Teil 2)

Kunde:

- Softwarehaus und die betroffenen Kunden

Problem:

- Es existiert eine Spalte SPARE1 die als VARCHAR2 definiert wurde. Der Inhalt soll aber korrekt in ein Datum für Abfragen umgewandelt werden.
- Klingt doch sehr einfach! -- Oder?

5. Datumsproblematik

Problemfall:

Vertikal partitionierte Tabellen mit einer gemeinschaftlich genutzten Tabelle.

Daher keine Definition von Date oder Check auf der Spalte möglich.



5. Datumsproblematik

ORIGINAL

10.12.2011
10.Juni.2011
1.Juni.2011
1.8.2011
1.08.2011
1.09.2011
1.3.11
21.3.011
21. 3.2011
21. Jul. 2011
21. J u l. 2011
21. 07. 02
23. 07
07.22
1.07.22
7/8/15
1-3/15
1-5-15
1-AUG-15
1-AUG -15
1-November -15
Lieferung 3 Wochen
Ende Januar

5. Datumsproblematik

Demo

5. Datumsproblematik

- to_date erkennt automatisch alle von Oracle möglichen Datumsangaben
- Fehler in dem Feld führen zum Abbruch des SQL-Statements
- Eine Funktion wird benötigt

```
CREATE OR REPLACE FUNCTION valid_date( p_str IN varchar2 ) RETURN date
DETERMINISTIC IS
BEGIN
    EXECUTE IMMEDIATE 'ALTER SESSION SET NLS_DATE_FORMAT = ''DD.MM.YY HH24:MI:SS'' ' ;
    EXECUTE IMMEDIATE 'ALTER SESSION SET NLS_DATE_LANGUAGE = ''GERMAN'' ' ;
    RETURN to_date( replace(p_str,' ',''));
EXCEPTION
    WHEN others THEN
        RETURN null;
END;
/
```

5. Datumsproblematik

ORIGINAL	KORRIGIERT
10.12.2011	10.12.11 00:00:00
10.Juni.2011	10.06.11 00:00:00
1.Juni.2011	01.06.11 00:00:00
1.8.2011	01.08.11 00:00:00
1.08.2011	01.08.11 00:00:00
1.09.2011	01.09.11 00:00:00
1.3.11	01.03.11 00:00:00
21.3.011	21.03.11 00:00:00
21. 3.2011	21.03.11 00:00:00
21. Jul. 2011	21.07.11 00:00:00
21. J u l. 2011	21.07.11 00:00:00
21. 07. 02	21.07.02 00:00:00
23. 07	
07.22	
1.07.22	01.07.22 00:00:00
7/8/15	07.08.15 00:00:00
1-3/15	01.03.15 00:00:00
1-5-15	01.05.15 00:00:00
1-AUG-15	01.08.15 00:00:00
1-AUG -15	01.08.15 00:00:00
1-November -15	01.11.15 00:00:00
Lieferung 3 Wochen	
Ende Januar	

5. Datumsproblematik

- Der Index kann erheblich verkleinert werden
- Null-Values werden nicht im Index eingetragen

```
Create index feld1_i90 on x1 (feld1);  
Create index feld2_i90 on x1 (valid_date(feld2));
```

```
select segment_name, bytes/1024/1024 mb  
  from dba_segments  
 where segment_name in ('FELD1_I90', 'FELD2_I90')
```

SEGMENT_NAME	MB
FELD1_I90	104.00
FELD2_I90	6.00

6. Schreibfehler in Feldern erkennen

Kunde:

- Pharma

Problem:

- Gibt es eine Möglichkeit, wenn auch begrenzt, Feldinhalte auf Schreibfehler zu prüfen?

6. Schreibfehler in Feldern erkennen

- **Woran erkennt man einen Schreibfehler?**
 - Könnten Buchstaben vertauscht sein (z.B. „udn“ = „und“, „aslo“ = „also“)
 - Sind Buchstaben hinzugefügt worden (z.B. „Entschuldigung“ = „Entschuldigungen“)
 - Sind andere Buchstaben genutzt worden (z.B. „Zirone“ = „Cirone“)
 - Wenige Buchstaben fehlerhaft
- **Welche Methoden können von der Datenbank genutzt werden?**
 - Oracle Text
 - Vergleiche mit Like
 - Vergleiche mit Soundex
 - Das Package UTL_MATCH

6. Schreibfehler in Feldern erkennen

Demo

6. Schreibfehler in Feldern erkennen

Beispiel für Schreibfehlererkennung mit `utl_match.edit_distance`.

Diese Funktion liefert die maximale Anzahl an Änderungsoperationen, damit die Werte identisch sind (0 = Volle Übereinstimmung)

```
select Name, soundex(Name) mit_soundex,  
       utl_match.edit_distance (Name, 'Meditossin') edit_distance  
from Medikament  
where utl_match.edit_distance (Name, 'Meditossin') between 0 and 2;
```

7. Suchen von ähnlich klingenden Inhalten

Kunde:

- Pharma
- In ähnlicher Form auch bei vielen anderen Kunden.

Problem:

- Wie können ähnliche Inhalte (z.B. Namen) gefunden werden?
- Wie kann damit eine Black-List aufgebaut werden?

7. Suchen von ähnlich klingenden Inhalten

- Welche Möglichkeiten gibt es?
 - Oracle Text
 - Vergleiche mit Like
 - Vergleiche mit Soundex
 - Regular Expression
 - **UTIL_MATCH.JARO_WINKLER_SIMILARITY**
 - Extra Packages (z.B. Kölner Phonetik)

7. Suchen von ähnlich klingenden Inhalten

- **Vorteile** SOUNDEX
 - Werte können mit einem Index versehen werden
 - Groß- und Kleinschreibung wird ignoriert
 - Vokale werden ignoriert (a, i, o, u, e und zusätzlich noch h, w, y)
 - Funktioniert gut für die englischen Sprache; weniger gut mit der deutschen Sprache
 - Braucht keinen Vergleichswert
- **Nachteile** SOUNDEX
 - Erster Buchstabe muss übereinstimmen
 - Abbruch der Generierung sobald die dritte Zahl ermittelt worden ist
 - Kann nur „gleich“ oder „nicht gleich“ bei einem Vergleich liefern

7. Suchen von ähnlich klingenden Inhalten

- **Vorteile** `UTL_MATCH.JARO_WINKLER_SIMILARITY`
 - Groß- und Kleinschreibung wird nicht ignoriert
 - Der Grad der Übereinstimmung wird in Prozent ermittelt
 - Auch der erste Buchstabe kann unterschiedlich sein
 - Sehr variables Suchraster
- **Nachteile** `UTL_MATCH.JARO_WINKLER_SIMILARITY`
 - Werte können nicht mit einem Index versehen werden
 - Braucht einen Vergleichswert

7. Suchen von ähnlich klingenden Inhalten

Demo

Fragen und Antworten

To be continued