Oracle Text – AllesTextOderWas?

Benedikt Nahlovsky
Managing Partner / Database Technology

Performing Databases GmbH
Wiesauer Straße 27
D – 95666 Mitterteich

Einleitung

Seit der Version 7 bietet Oracle die Möglichkeit der Volltextsuche. Die frühen Optionen mussten noch separat installiert werden und in der Oracle -Version 8i war die Textsuche in das kostenpflichtige Zusatzmodul interMedia integriert. Seit Version 9i ist Oracle Text jedoch fester Bestandteil der Datenbank, auch in der Express Edition. Dies kann man leicht nachprüfen, indem man nach dem User ctxsys sucht:

```
[oracle@oravm11gR2 ~]$ (ORAVM11) sqlplus perfdbtext/pdb

SQL*Plus: Release 11.2.0.4.0 Production on Thu Oct 23 17:36:02 2014

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

Connected to:

Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit
Production

With the Partitioning, Automatic Storage Management, OLAP, Data Mining and
Real Application Testing options

SQL> set lines 1000
SQL> set pages 100
SQL>
SQL> select * from all_users where username='CTXSYS';
no rows selected
SQL>
```

Installation von Oracle Text

Anlegen des CTXSYS Schemas - die Oracle Text Komponenten werden in das Schema CTXSYS installiert:

```
SQL> conn / as sysdba

Connected.

SQL> @?/ctx/admin/catctx.sql ctxsys SYSAUX TEMP NOLOCK

...creating user CTXSYS
```

Nach der Installation empfiehlt sich die Standard Spracheinstellungen, wie zum Beispiel einen DEFAULT_LEXER, in die Datenbank zu laden. Andernfalls können Oracle Text Aufrufe mit einem DRG-10700: preference does not exist: CTXSYS.DEFAULT_LEXER fehlschlagen.

```
SQL> conn ctxsys/ctxsys

Connected.

SQL> @?/ctx/admin/defaults/dr0defin.sql "GERMAN";
```

Überprüfung der Installation

Im Anschluss kann mit dem folgenden SQL Statement überprüft werden, ob das Datenbankfeature korrekt installiert wurde.

```
SQL> select comp_name, status, version from dba_registry where
comp_id=,CONTEXT';

COMP_NAME                                STATUS      VERSION

---------------------------------------- ----------- -------------

Oracle Text                              VALID       11.2.0.4.0
```

Die Basisfunktionen kann man ohne zusätzlichen Rechte nutzen, für die Anpassung der Sucheinstellungen braucht man jedoch mindestens das Execute-Recht auf das wichtigste Package des Schemas ctxsys: cx_ddl oder die Rolle ctxapp.

```
SQL> conn / as sysdba

Connected.

SQL> grant execute on ctx_ddl to perfdbtext;

Grant succeeded.

SQL> grant ctxapp to perfdbtext;

Grant succeeded.

SQL>
```

Wie funktioniert Oracle Text?

1. Die Dokumente werden in einen so genannten Datastore eingelesen. Die zu indizierenden Texte können dabei entweder in CLOB-, VARCHAR2- oder XMLTYPE-Spalten einer Tabelle in der Datenbank liegen (direct_datastore), im Filesystem des Datenbankservers (file_datastore) oder im Inter- bzw. Intranet (url_datastore). Es besteht sogar die Möglichkeit, die Texte über eine selbst definierte Prozedur direkt vor der Indizierung zusammenzustellen (user_datastore).

2. Im 2. Schritt werden die Objekte im Bedarfsfall gefiltert. Das ist nur dann nötig, wenn es sich um binäre Files, wie Word-Dokumente oder PDF-Dateien handelt. Text-, HTML- und XML-Dateien müssen nicht gefiltert werden. Oracle erkennt über 150 Formate automatisch.

3. Der Sectioner kann HTML- oder XML-Dokumente anhand von Tags (z.B. <H1>...</H1> in HTML oder <Produktbeschreibung>...</Produktbeschreibung> in XML) in einzelne Abschnitte aufteilen.

4. Der Lexer extrahiert alle relevanten Wörter aus dem Text. Interpunktions- und Sonderzeichen werden entfernt. Bei diesem Schritt kann man unter anderem einstellen,

   was als Trennzeichen gewertet bzw. ignoriert werden soll (Leerzeichen, Unterstriche etc.) ob Groß- und Kleinschreibung beibehalten werden soll,
   ob zusammengesetzte Worte in Ihre Einzelteile zerlegt werden sollen, etc.

5. Beim Indizierungsprozess wird aus den gesammelten Wörtern ein invertierter Index erzeugt. Jedem Wort wird dabei eine Liste seiner Fundstellen zugeordnet.

   Artikel, Konjunktionen, Präpositionen und Hilfsverben etc., bei Oracle Text Stopwörter genannt, werden nicht indiziert.
   Die Einstellungen der sog. wordlist legen fest, welche grammatikalischen Regeln verwendet werden sollen, damit bei der Textsuche auch Beugungsformen des gesuchten Verbs oder Wörter mit ähnlichem Stamm erkannt werden.

Volltextsuche in Textspalten

Wir erstellen die Tabelle und erzeugen einen Index ohne zusätzlichen Parameter:

```
SQL> CREATE TABLE TEXTTAB (

  ID NUMBER NOT NULL

, TEXT VARCHAR2(800)

, CONSTRAINT TEXTTAB_PK PRIMARY KEY(ID) ENABLE );

SQL> CREATE INDEX text_idx ON texttab(text) INDEXTYPE IS ctxsys.context;

Index created.

SQL>
```

Die Suche in einem Context-Index wird über das Schlüsselwort CONTAINS durchgeführt:

```
SQL> SELECT spaltenliste FROM tabelle WHERE CONTAINS(index_spalte,
'<suchbegriff>')>0;
```

Die wichtigsten Suchmöglichkeiten:

1. Einfache Suche nach Wörtern, z.B.:

```
SQL> SELECT * FROM texttab WHERE CONTAINS(text, 'Landkreis') > 0;

ID TEXT
-------------------------------------------------------------------------------

5 Der Landkreis Schwandorf liegt in Bayern.
6 Der Landkreis Regensburg ist größer als Schwandorf.
9 Mitterteich liegt im Landkreis Tirschenreuth.
```

2. Suche nach Wort-Kombinationen oder -Alternativen mit den Booleschen Operatoren „AND“ und „OR“:

```
SQL> SELECT * FROM texttab WHERE CONTAINS(text, 'Landkreis AND Bayern') >
0;

ID TEXT
-------------------------------------------------------------------------------

5 Der Landkreis Schwandorf liegt in Bayern.

SQL> SELECT * FROM texttab WHERE CONTAINS(text, 'Landkreis OR Bayern') > 0;

ID TEXT
-------------------------------------------------------------------------------

2 Die Landeshauptstadt von Bayern ist Munchen.
5 Der Landkreis Schwandorf liegt in Bayern.
6 Der Landkreis Regensburg ist größer als Schwandorf.
9 Mitterteich liegt im Landkreis Tirschenreuth.
```

3. Suche nach ähnlich geschriebenen Wörtern:
Wenn man den Operator „?“ vor das gesuchte Wort stellt, kann man auch Wörter mit Rechtschreibfehlern oder Buchstabendrehern finden:

```
SQL> SELECT * FROM texttab WHERE CONTAINS(text, '?Baiern') > 0;

ID TEXT

-------------------------------------------------------------------------------

2 Die Landeshauptstadt von Bayern ist Munchen.
5 Der Landkreis Schwandorf liegt in Bayern.
```

4. Suche mit Wildcards: „%“ für kein oder beliebig viele Zeichen und „_“ für genau 1 Zeichen:

```
SQL> SELECT * FROM texttab WHERE CONTAINS(text, '%dor_') > 0;

ID TEXT
-------------------------------------------------------------------------------

5 Der Landkreis Schwandorf liegt in Bayern.
6 Der Landkreis Regensburg ist größer als Schwandorf.
```

5. Suche nach Ausdrücken, die denselben Wortstamm haben wie das Suchwort oder mit dem Suchwort zusammengesetzte Worte bilden, mit dem Operator „$“:

```
SQL> SELECT * FROM texttab WHERE CONTAINS(text, '$liegen') > 0;

ID TEXT
--------------------------------------------------------------------------------
5 Der Landkreis Schwandorf liegt in Bayern.
9 Mitterteich liegt im Landkreis Tirschenreuth.
```

Synchronisation und Optimierung des Indexes

In früheren Versionen von Oracle Text musste der Index nach DML-Operationen manuell oder über DBMS_JOB bzw. DBMS_SCHEDULER neu aufgebaut werden.
Ab Version 10g hat man auch hier mehrere Möglichkeiten.

1. Manuell:

```
SQL> INSERT INTO texttab VALUES(11, 'New York ist eine Stadt an der
Ostseeküste in den USA');

1 row created.

SQL> SELECT * FROM texttab WHERE CONTAINS(text, 'USA') > 0;

no rows selected

SQL> exec ctx_ddl.sync_index('text_idx');

PL/SQL procedure successfully completed.

SQL> SELECT * FROM texttab WHERE CONTAINS(text, 'USA') > 0;

ID TEXT
--------------------------------------------------------------------------------
11 New York ist eine Stadt an der Ostküste in den USA

SQL>
```

2. Über einen Job in regelmäßigen Intervallen. Dazu gab es in der Oracle DB Version 9 ein kleineres Skript namens drjobdml.sql, dass den Namen des Indexes und das Intervall in Minuten per Austauschvariable übernimmt:
Der Index wird nun alle 60 Minuten synchronisiert.

```
SQL> set define on DECLARE
1 job NUMBER; BEGIN
2 dbms_job.submit(job, 'ctx_ddl.sync_index(''text_id'');', interval =>
'SYSDATE +60/1440');
3 commit; END;
4/
```

3. Automatisch in regelmäßigen Intervallen
Diese Option kann man beim Anlegen des Indexes oder nachträglich einrichten. Man braucht dazu das Create-Job Recht.

Synchronisation jeden Tag um Mitternacht

```
SQL> CREATE INDEX text_idx ON texttab(text) INDEXTYPE IS CTXSYS.CONTEXT
PARAMETERS ('SYNC (EVERY "TRUNC(sysdate)+1/24")');

Index created.
```

oder nachträglich (hierbei werden nur die Metadaten des Index verändert, nicht die Struktur):

```
SQL> ALTER INDEX text_idx REBUILD PARAMETERS (REPLACE METADATA 'SYNC (EVERY
"TRUNC(sysdate)+1/24")';

Index altered.
```

4. Automatisch nach jedem Commit (ab Oracle 10g). Dies ist nur dann sinnvoll, wenn selten DML-Aktionen stattfinden, weil der Index sonst unnötig fragmentiert:

```
SQL> CREATE INDEX text_id ON texttab(text) INDEXTYPE IS ctxsys.context
PARAMETERS ('SYNC (ON COMMIT)');

Index created.
```

bzw.:

```
SQL> ALTER INDEX text_id REBUILD PARAMETERS ('REPLACE METADATA SYNC (ON
COMMIT)');

Index altered.
```

5. Automatisch nach jeder Transaktion (ab Oracle 10g)
Damit werden Änderungen in den Texten sofort registriert (in der Tabelle dr$unindexed im Schema ctxsys), die Synchronisation muss aber zusätzlich erfolgen, damit die Anfrageperformance mit wachsender Größe dieser Tabelle nicht in den Keller geht:

```
SQL> CREATE INDEX text_id ON texttab(text) INDEXTYPE IS ctxsys.context
PARAMETERS (,TRANSACTIONAL');

Index altered.
```

bzw.:

```
SQL> ALTER INDEX text_id REBUILD PARAMETERS ('REPLACE METADATA
TRANSACTIONAL');

Index altered.
```

Eine Optimierung des Indexes ist über die Prozedur CTX_DDL.OPTIMIZER_INDEX möglich. Je nach Änderungshäufigkeit sollte man den Index über einen Job in regelmäßigen Abständen optimieren, z.B. einmal pro Tag mit:

```
SQL> BEGIN

1 ctx_ddl.optimize_index(index_name => 'text_idx', optlevel => 'FULL');

2 END;
```

Kontaktadresse:
Benedikt Nahlovsky
Performing Databases GmbH
Wiesauer Straße 27
D-95666 Mitterteich

Telefon: +49 (0) 9633-631
Mobil: +49 (0) 170-7373326
Fax: +49 (0) 9633-4199
E-Mail benedikt.nahlovsky@performing-db.com
Internet: http://www.performing-databases.com