Oracle Text - Ein vergessenes Feature in der Oracle Datenbank

Stephan La Rocca
PITSS GmbH
Bielefeld

Schlüsselworte
Oracle Text, Datenbank Edition, PL/SQL-Entwicklung, Datenmodellierung,

Einleitung
Denken Sie an Content Management Systeme, Wikis, oder die Verwaltung von Artikeln und Dokumenten ist ihnen sicherlich sofort klar, dass mit der Volltext Suche von Oracle ein Benefit erreicht werden kann. Aber wussten Sie z.B. dass Oracle Text in jeder Datenbank-Edition kostenfrei enthalten ist und sogar für einfache Anwendungen, wie z.B. eine Adressverwaltung sich trefflich einsetzen lassen.
Oft sind es nur die kleinen Hürden der Installation und Administration, die es einmal zu nehmen gilt, bevor die eigene Applikation auf Grund der vielzähligen neuen Möglichkeiten eine völlig andere User Experience erfährt.

Installation
Die erste Hürde, die es in der Regel zu nehmen gilt, ist die Frage, ob das Feature schon in meiner Datenbank installiert ist, und wenn nicht, was dann zu tun ist. Dazu gibt es bereits im Netz einen hilfreichen Artikel von Ulrike Schwinn in dem gemeinsamen Blog mit Carsten Czarksi (http://oracle-text-de.blogspot.de/)

```
SQL> SELECT comp_name, status, substr(version,1,10) as version

     FROM dba_registry WHERE comp_id = 'CONTEXT';
```

Sollte die Komponente fehlen, kann sich recht einfach über das Skript ctx/admin/catctx.sql nachinstalliert werden. Die Nutzung dieser Komponente ist unabhängig von der Datenbank-Edition und selbst in der XE-Edition ist die Oracle Text-Komponente schon perse vorhanden.

Einsatzgebiete
Für die Komponente Oracle Text gibt es vier große Einsatzgebiete, die es in zunächst zu unterscheiden gilt:
Dokumenten-Management-Systeme: Bei dieser Art von Anwendungen stehen HTML, PDF, Word oder andere Arten von Dokumenten und deren Verwaltung, insbesondere die performante Volltext-Recherche (Wiki, etc.) im Vordergrund. (*CONTEXT*)
Shop-Lösungen: Bei dieser Art von Anwendungen sind Artikel und deren Bezeichnungen und Beschreibungen parallel zu Preisen, Verfügbarkeiten, etc. zu pflegen. (*CTXCAT*)
Klassifizierungs-Systeme:Diese Anwendungen haben die Aufgabe, effizient und sicher eingehende Dokumente (z.B. Nachrichtenticker, News-Feeds, etc) zu klassifizieren. (*CTXRULE*)
XML-Management: Speziell die Verwendung von XML-Doumenten vereinen die Nutzung strukturelle Informationen und die Notwendigkeit einer Volltextsuche. (*CTXXPATH*)

Je nach Einschätzung der eigenen Applikation in eine der vier Kategorien, bietet sich auch die Verwendung eines unterschiedlichen Index-Typen an. Diese befinden sich kursiv in den Klammern jeweils am Ende der obigen Auflistung.

Verwendungsbeispiel
In einem wachsenden potentiellen Anwendungsgebiet und der Tatsache, dass dieser Typ aus meiner persönlichen Wahrnehmung etwas in Vergessenheit geraten ist, soll der CTXCAT-Index-Typ etwas näher beleuchtet werden.
Das typische Retrieval in einem WebShop zeigt sich wie folgt:

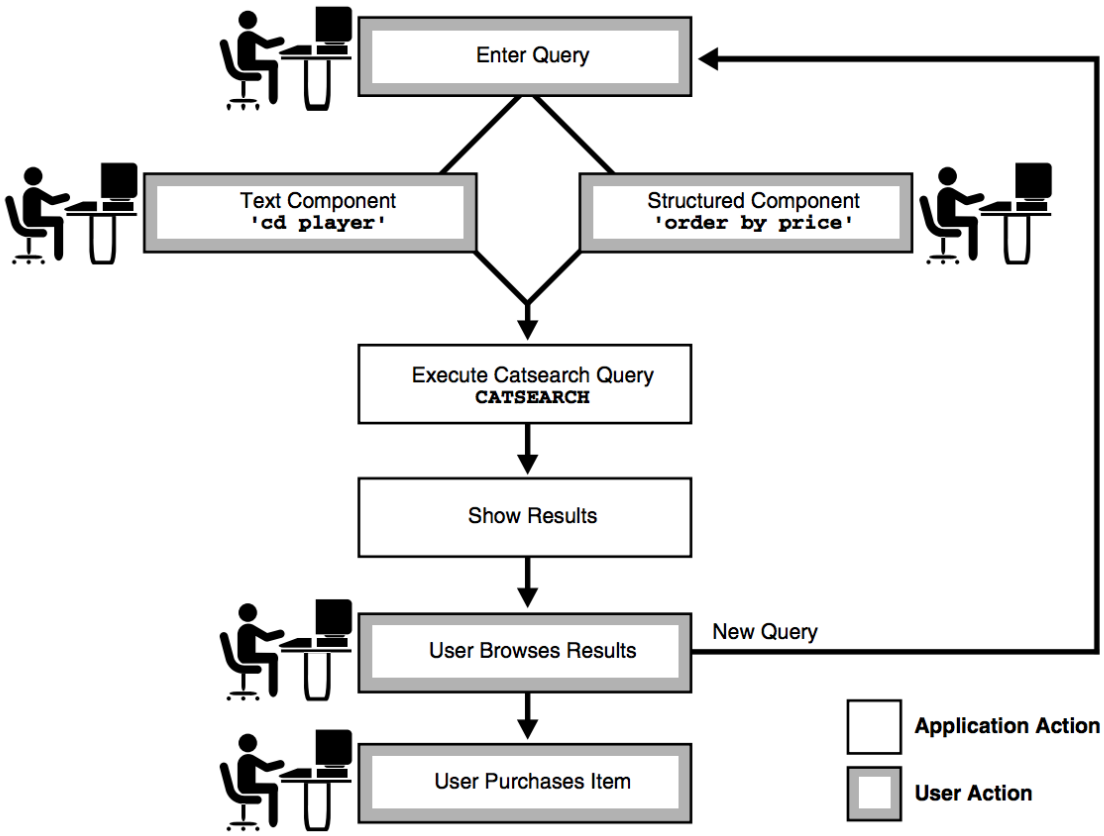

Abb. 1: Flussdiagramm einer Shop-Lösung (Quelle Oracle Text Application Developer Guide)

Die Abfrage an das Datenmodell, die möglichst performant beantwortet werden soll, beinhaltet neben unstrukturierten Informationen (irgendwo in der Artikelbezeichnung oder Beschreibung sollen die Wörter CD und Player vorkommen) auch strukturelle Informationen (sortiert nach Preis).
Der dazu präferierte Index-Typ ist der CTXCAT-INDEX. Dieser Indextyp ermöglich die Kombination von Spalten aus einer Tabelle mit dem Volltext-Index auf weiteren Attributen. Strukturiert wird der Index in einem Index-Set für eine Tabelle. Ein Index-Set besteht aus mehreren Subindexen, die klassisch die Spalten indizieren, die später für die Abfrage oder Sortierung notwendig werden.

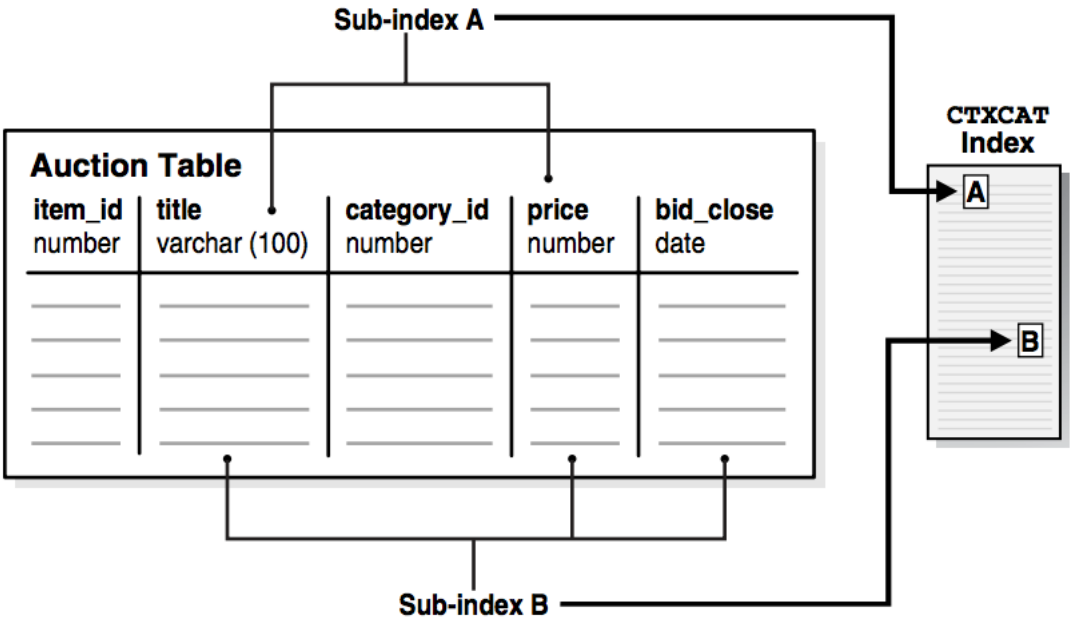

Abb. 1: Tabellenschema und CTXCAT-INDEX (Quelle Oracle Text Application Developer Guide)

In dem obigen Beispiel wird ein Sub-Index A auf der Spalte „price“ erstellt, da diese Information für die Sortierung genutzt werden soll. Der Sub-Index B schließt die Spalte „bid-close“ für weitere Filterkriterien ein. Für die Umsetzung muss in einem ersten Schritt zunächst das Index-Set für die Tabelle angelegt werden:

```
begin

ctx_ddl.create_index_set('auction_iset');

end;
```

Anschließend können die Sub-Indizes angelegt werden, die für die Applikation sinnvoll erscheinen:

```
begin

ctx_ddl.add_index('auction_iset','price'); /* sub-index A */

ctx_ddl.add_index('auction_iset','price, bid_close'); /* sub-index B */

end;
```

Abschließend wird dieses Index-Set zusammen mit der Information über die zu indizierende Spalte auf die Tabelle gelegt:

```
CREATE INDEX auction_titlex ON AUCTION(title)
```

```
 INDEXTYPE IS CTXSYS.CTXCAT
 PARAMETERS ('index set auction_iset');
```

Bei dem Retrieval-auf diese Tabelle kommt dann die Funktion Catsearch zum Einsatz, die die Where-Clause für die Kombination beschreibt:

```
SELECT FROM auction WHERE CATSEARCH(

 title, 'camera','price = 100

 ORDER BY bid_close')> 0;

SELECT FROM auction
WHERE CATSEARCH(

 title, 'camera','order by price, bid_close')> 0;
```

Anwendung in einer Adressverwaltung

Neben dem obigen Beispiel für ein Shop-System ergeben sich aber noch ganz andere Möglichkeiten für die Verwendung dieser Komponente.
In einer Adressverwaltung beispielsweise können sie sich noch so viele Regeln auferlegen, wie Firmen, Vereine, Privatpersonen, etc struktutiert erfasst werden sollen. Wenn ein anderer Mitarbeiter eine Person sucht, steht sie dann doch immer wieder vor den Fragen: Steht meine gesuchte Person im Namenfeld, oder doch die Firmenbezeichnung? Ist Herr Müller nur der Ansprechpartner oder was steht im Firmenzusatz?
Um diesem Wirrwarr zumindest bei der Suche zu umgehen, bietet sich folgendes Verfahren an:

1.) Strukturieren Sie die Adressdatensätze, Kontaktdaten, etc. nach dem besten Verständnis der normalisierung und den gängigen DIN-Vorschriften.
2.) Packen Sie eine „Pseudespalte“ (am besten vom Datentyp CLOB) an die Basis-Tabelle des Datenmodells.
3.) Befüllen Sie diese Spalte über einen Trigger mit einem konkatinierten (vielleicht in einer JSON-Notation) String mit allen Attributen ihres Datenmodells (Name1, Name2, Vorname, Firma, Firmenzusatz, etc,) nach denen der Anwender später suchen können soll.
4.) Erstellen sie Sub-Indizes auf allen Spalten, nach denen typischerweise gesucht wird: PLZ, Ort, Straße,
5.) Erstellen Sie ein Index-Set über diese Subindizes zusammen mit der erstellten Pseudo-Spalte.

Somit haben Sie die Basis geschaffen für eine Anwendung, die tolerant über das gesamte Datenmodell suchen kann.
Mit etwas dynamischer Programmierung bekommen Sie auf diese Art und Weise sogar ein datenmodellinternes Google realisiert, welches über alle varchar2-Spalten Ihres Modells suchen kann.

Administration

Im Gegensatz zum CONTEXT-Index ist der CTXCAT-INDEX transaktional, d.h. die Änderungen an den Datensätzen werden synchron direkt in den Index überführt, so dass ein Synchronisieren oder Neuaufbau des Index nicht notwendig ist.
Hilfreiche Tools und Auswertungen zeigt die Enterprise Manager Console in der Rubrik Text-Manager.

Kontaktadresse:
Stephan La Rocca
PITSS GmbH
Otto-Brenner-Str. 209
D-33604 Bielefeld

Telefon:	+49 (0) 521-546 795-07
Fax:	+49 (0) 521-546 795-01
E-Mail	slarocca@pitss.de
Internet:	www.pitss.de