



Volltextsuche in XML-Dokumenten mit Oracle

Carsten Czarski,
Nina Neuwirth
Oracle Deutschland

ORACLE®

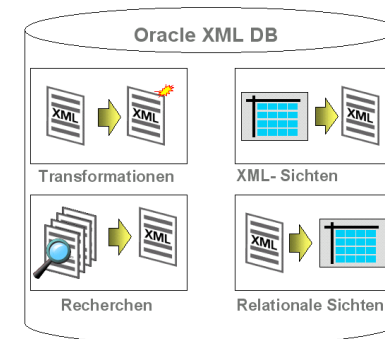


Agenda

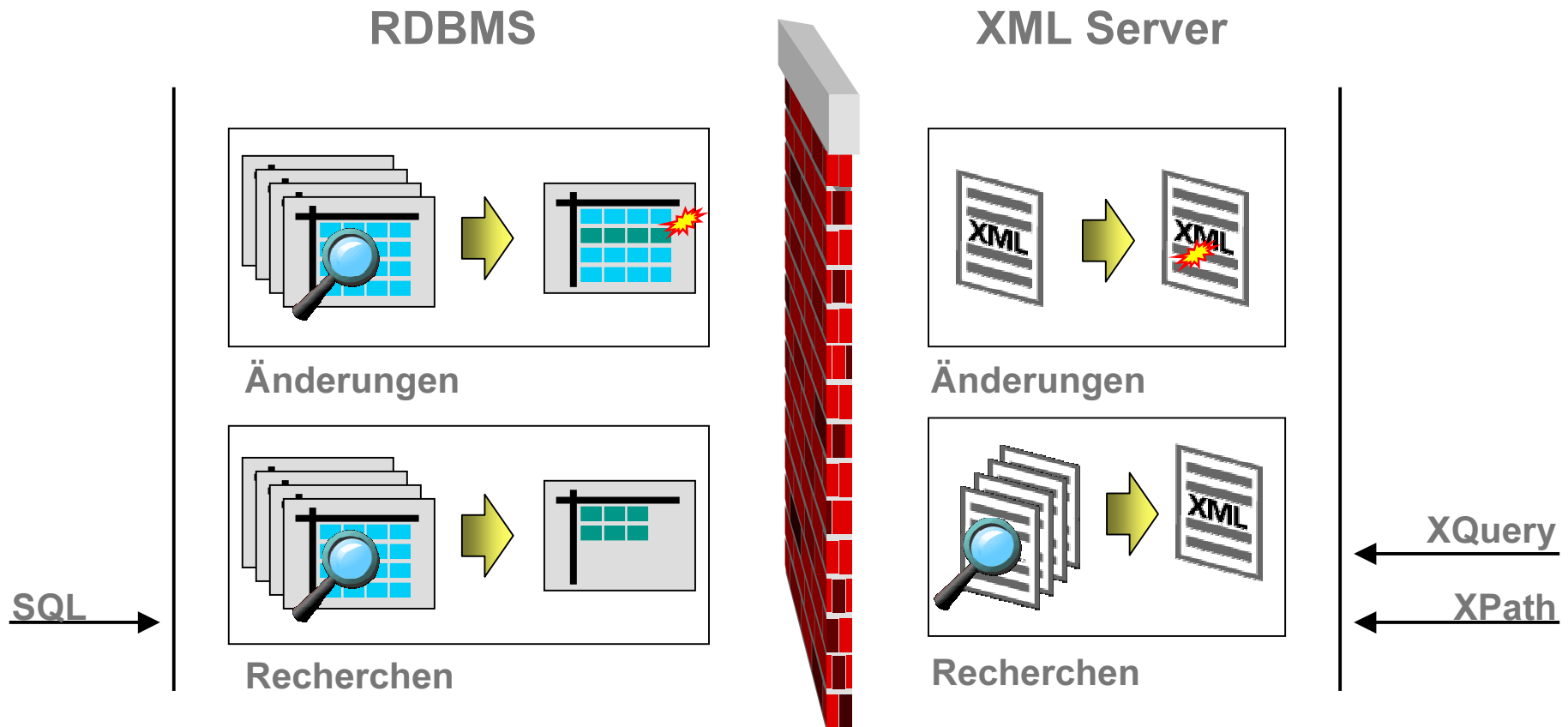
- Oracle XML DB: Kurzaufriß
- Volltextsuche in XML-Dokumenten
- Tipps und Tricks

Oracle XML DB Kurzprofil

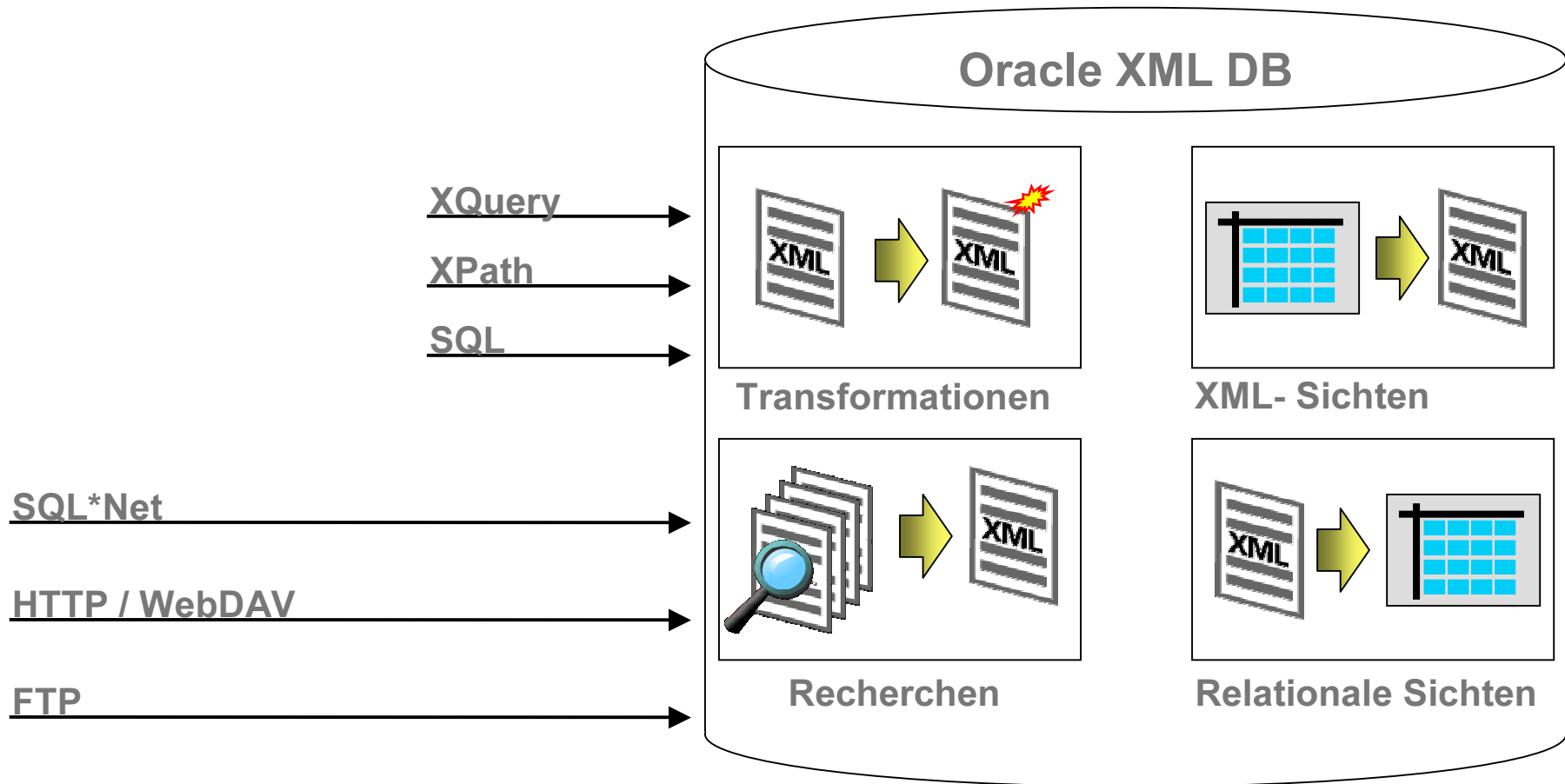
- XML und SQL in einer Datenbank
 - Nahtlose Integration
- Standardkonform ...
 - XML/SQL (SQL:2003)
 - XQuery
 - XML Schema, DOM
- Verfügbar ab Oracle 9i Release 2
- Alle Datenbankeditionen



Oracle XML DB: zwei Welten ...



... wachsen zusammen!



XML DB: Zugriffe

The image shows two overlapping browser windows. The background window is Mozilla Firefox, displaying an index page with links for [home/](#), [i/](#), [public/](#), and [sys/](#). The foreground window is Microsoft Internet Explorer, displaying XML data retrieved from a database. The XML content is as follows:

```
<?xml version="1.0" encoding="iso-8859-1" ?>
- <query >
  <titel xmlns="http://www.oracle.com/aktie/nachrichten.xsd">Wall Street er6ffnet behauptet,
  Oracle-Zahlen an der NASDAQ im Fokus</titel>
  <titel xmlns="http://www.oracle.com/aktie/nachrichten.xsd">Oracle sieht hohen Zuspruch f6r
  Angebot bei PeopleSoft-Aktion6ren</titel>
  <titel xmlns="http://www.oracle.com/aktie/nachrichten.xsd">Oracle beruhigt PeopleSoft
  Kunden</titel>
  <titel xmlns="http://www.oracle.com/aktie/nachrichten.xsd">Oracle kooperiert mit Nokia und
  Alcatel</titel>
</query >
```

Beispiel: Börsennachrichten

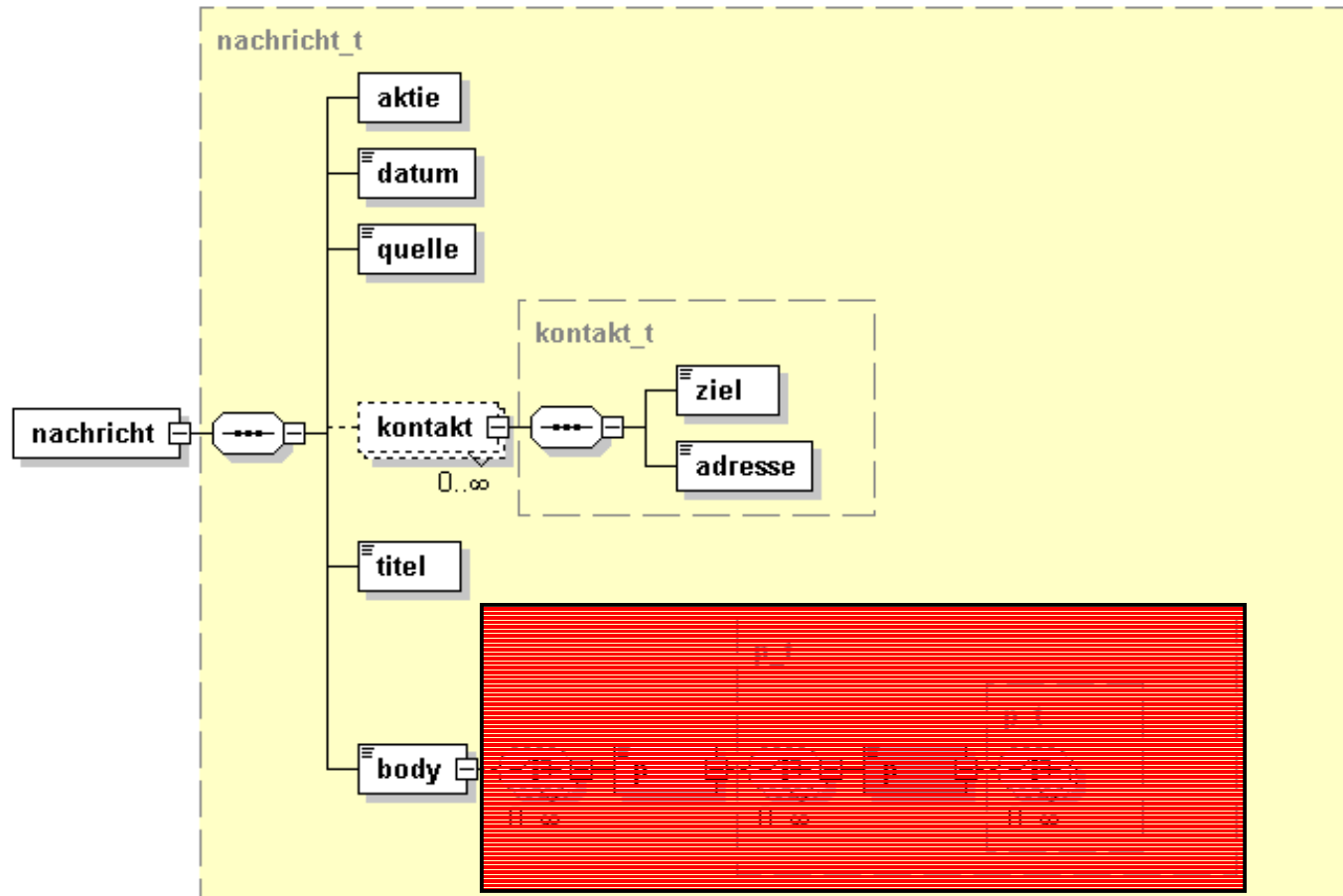
```
<?xml version="1.0" encoding="utf-8" ?>
- <nachricht xmlns="http://www.oracle.com/aktie/nachrichten.xsd" typ="Unternehmensmeldung"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.oracle.com/aktie/nachrichten.xsd
  http://www.oracle.com/aktie/nachrichten.xsd">
  <aktie wkn="888811" name="Oracle Corporation" reuters="ORCL" branche="Software" />
  <datum>2003-02-18T10:23:00</datum>
  <quelle>Finanzen.net</quelle>
- <kontakt art="email">
  <ziel>Oracle</ziel>
  <adresse>investor_us@oracle.com</adresse>
</kontakt>
- <kontakt art="email">
  <ziel>Finanzen.net</ziel>
  <adresse>fred.redakteur@finanzen.net</adresse>
</kontakt>
  <titel>Oracle kooperiert mit Nokia und Alcatel</titel>
- <body>
  <p>Der Softwarekonzern Oracle meldete am Dienstag eine Kooperation mit den
  Telekommunikationsausrüstern Nokia (Finnland) und Alcatel (Frankreich).</p>
  <p>Auf diese Weise will Oracle die Verbreitung eigener Software-Lösungen vorantreiben. Bislang
  hält der Softwarekonzern Microsoft das Geschäft rund um die Anwendungsgebiete der "Outlook
  Office Software" fest in der Hand. Microsoft hat bereits seit einiger Zeit damit begonnen,
  Kooperationen im Mobilfunkbereich zu schließen.</p>
  <p>Im Zuge der Zusammenarbeit mit Nokia und Alcatel wird der finnische Konzern die Email-
  Software, das Voicemail-Modul sowie das Kalender- und File-Management-System auf seinen
  Terminals installieren. Mit Alcatel ist eine verstärkte Konzentration auf den
  Geschäftskundenbereich geplant.</p>
  <p>Die Aktie von Oracle schloss gestern an der NASDAQ mit einem Gewinn von 1,39 Prozent bei
  11,70 Dollar. Das Papier von Nokia verliert in Amsterdam derzeit 2,03 Prozent auf 13,01 Euro. Der
  Anteilsschein von Alcatel gibt in Paris aktuell um 1,35 Prozent auf 7,30 Euro nach.</p>
  </body>
</nachricht>
```



Beispiel: Börsennachrichten Überlegungen ...

- Strukturierte Informationen
 - (WKN, Reuters-Code, Quelle, Kontakte, Datum)
 - Zugriff auf Elemente im Vordergrund
 - Kombination mit bspw. Kursdatenbank
 - Strukturierte Speicherung
- Unstrukturierte Informationen
 - Zugriff auf den vollständigen Text im Vordergrund
 - Volltextsuche
 - Unstrukturierte Speicherung

Beispiel: Börsennachrichten Beschreibung als XML Schema



Beispiel Börsennachricht

Registrierung des XML Schemas

```
C:\WINNT\System32\cmd.exe
Session wurde geändert.
Abgelaufen: 00:00:00.00
09:51:53 SQL>
09:51:53 SQL> prompt registering new xml schema
registering new xml schema
09:51:53 SQL>
09:51:53 SQL> begin
09:51:53      2      dbms_xmlschema.registeruri
09:51:53      3      (
09:51:53      4          schemaur1 => 'http://www.oracle.com/aktie/nachrichten.xsd'
09:51:53      5          ,schemadocuri => '/public/DOAG/nachrichten_annotiert.xsd'
09:51:53      6      );
09:51:53      7      end;
09:51:53      8      /

PL/SQL-Prozedur wurde erfolgreich abgeschlossen.

Abgelaufen: 00:00:03.03
09:51:56 SQL> show err
Keine Fehler.
09:51:56 SQL>
09:51:56 SQL> --exit
09:51:56 SQL>
```



Beispiel Börsennachrichten

- Neue Tabellen "manuell" anlegen
 - XMLTYPE als Datentyp

```
CREATE TABLE nachrichten_tab
(
  nr_id          number(10),
  nr_nachricht  xmltype
)
xmltype column nr_nachricht store as object relational
xmlschema "<schema-url>"
element "<root element>"
```



XML Dokumente laden – Referenz auf XML Schema

```
<nachricht  
  xmlns="http://www.oracle.com/aktie/nachrichten.xsd"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  typ="Unternehmensmeldung"  
  xsi:schemaLocation="http://www.oracle.com/aktie/nachrichten.xsd  
                      http://www.oracle.com/aktie/nachrichten.xsd"  
>
```

- Laden mit [FTP](#) und [WebDAV](#)
 - XML-Dokument muss das XML Schema referenzieren
- SQL und SQL*Loader



Schemavalidierung

- "leichte" Schemavalidierung
 - beim Einfügen
 - Strukturprüfung
 - Performancegründe
- vollständige Schemavalidierung
 - aktivierbar
 - Trigger
 - **`XMLTYPE.schemaValidate()`**



Zugriff mit SQL

Gib mir von allen Oracle-Nachrichten
das Datum und den Titel

```
select
  extractvalue
    (value(e), '/nachricht/datum') as datum
,extractvalue
  (value(e), '/nachricht/aktie/@name') as ag
,extractvalue
  (value(e), '/nachricht/titel') as titel
from nr_nachricht_tab e
where existsnode
(
  value(e), '/nachricht/aktie[@wkn="871460"]'
) = 1
```

Mit „Query Rewrite“

```
SELECT  
  extractvalue ...  
FROM nr_nachricht_tab e
```

Objektrelational gespeichertes
XML-Dokument

Query Rewrite

```
SELECT  
  e.xmldata. ...  
FROM nr_nachricht_tab e
```

nachricht:
aktie
datum
quelle
kontakte
titel
body

1

1

1

*

aktie:
wkn
name
branche
reuters

kontakte:
ziel
art
adresse

Oracle TEXT und XML Dokumente

- Text On Top (ToT)
 - XML Strukturen in einer Oracle TEXT Abfrage
- XML On Top (XoT)
 - Volltextsuche in einer XML Abfrage





Oracle TEXT Index auf XML Dokumente

- SECTION GROUPS
 - XML_SECTION_GROUP
 - AUTO_SECTION_GROUP
 - PATH_SECTION_GROUP
- Operatoren
 - WITHIN
 - INPATH / HASPATH



XML_SECTION_GROUP

- Prinzip:
 - Nur deklarierte Abschnitte werden indiziert
 - Keine vordefinierten Abschnitte
- CONTAINS-Operatoren
 - WITHIN
- Vorteile
 - Tendenziell kleinere Indizes
 - Kontrolle über zu indizierende Dokumentteile
- Nachteile
 - Relativ hoher Implementierungsaufwand
 - Keine XPath-ähnliche Syntax möglich



XML_SECTION_GROUP

```
begin
  ctx_ddl.create_section_group
  (
    group_name => 'nachrichten_sec_group';
    group_type => 'XML_SECTION_GROUP'
  );
end;

begin
  ctx_ddl.add_field_section
  (
    group_name      => 'nachrichten_sec_group',
    section_name    => 'titel',
    tag              => 'titel'
  );
end;
```



XML_SECTION_GROUP

```
begin
  ctx_ddl.add_zone_section(
    group_name => 'nachrichten_sec_group',
    section_name => 'sec_nachricht',
    tag => 'nachricht'
  );
end;

begin
  ctx_ddl.add_zone_section(
    group_name => 'nachrichten_sec_group',
    section_name => 'adresse',
    tag => 'adresse'
  );
end;
/
```



AUTO_SECTION_GROUP

- Prinzip:
 - Standardmäßig vollständige Indizierung
 - Deklaration von sog. "STOP"-Abschnitten möglich
- CONTAINS-Operatoren
 - WITHIN
- Vorteile
 - Implementierung einfacher
- Nachteile
 - Keine XPath-ähnliche Syntax möglich



AUTO_SECTION_GROUP

```
begin
  ctx_ddl.create_section_group
  (
    group_name => 'nachrichten_sec_group',
    group_type => 'AUTO_SECTION_GROUP'
  );
end;

begin
  ctx_ddl.add_stop_section
  (
    group_name => 'nachrichten_sec_group',
    tag         => 'adresse'
  );
end;
```



PATH_SECTION_GROUP

- Prinzip
 - Stets vollständige Indizierung des Dokumentes
- CONTAINS-Operatoren
 - WITHIN
 - INPATH und HASPATH
- Vorteile
 - XPath-Ähnliche Syntax möglich
 - Einfache Implementierung
- Nachteile
 - Keine Möglichkeit zum Ausschließen bestimmter Abschnitte
 - Indexumfang größer



PATH_SECTION_GROUP

- Ein Beispiel ...

```
begin
  ctx_ddl.create_section_group
  (
    group_name => 'nachrichten_sec_group'
    ,group_type => 'PATH_SECTION_GROUP'
  );
end;
```



Index erstellen ...

- Wie immer ...

```
create index idx_nachrichten_volltext
on nr_nachricht_tab (object_value)
indextype is CTXSYS.CONTEXT
parameters ('section group nachrichten_sec_group')
```



Abfrageoperator: WITHIN

- Unterstützt von allen "Section groups"
- Verschachtelungen möglich

```
select rowid from NR_NACHRICHT_TAB
where CONTAINS(
    object_value,
    'Kooperation within titel'
)>0;

select rowid from NR_NACHRICHT_TAB
where CONTAINS(
    object_value,
    '(Kooperation within titel) within body'
)>0;
```



Abfrageoperator: INPATH

- Unterstützt von PATH_SECTION_GROUP
- XPath-ähnliche Syntax

```
select rowid from NR_NACHRICHT_TAB
where CONTAINS(
    object_value,
    'Kooperation inpath (//titel)'
)>0;

select rowid from NR_NACHRICHT_TAB
where CONTAINS(
    object_value,
    'Kooperation inpath (//body/titel)'
)>0;
```



Abfrageoperator: HASPATH

- Unterstützt von PATH_SECTION_GROUP
- XPath-ähnliche Syntax
- Prüft die Existenz eines Pfades

```
select rowid from NR_NACHRICHT_TAB
where CONTAINS(
    object_value,
    'haspath (//[titel=Kooperation]) '
)>0;
```

```
select rowid from NR_NACHRICHT_TAB
where CONTAINS(
    object_value,
    'haspath (//body[titel=Kooperation]) '
)>0;
```



XML-Besonderheiten und INPATH / HASPATH

- XML Standards
 - Keine 100%-Unterstützung des XPath-Standards
 - ABER: XPath definiert keine Volltextsuche
 - Keine Unterstützung für benutzerdefinierte Entities
 - Keine XML-Namensräume



XPath-Volltextsuche

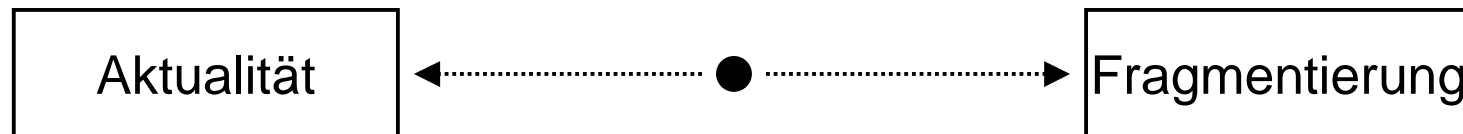
- XPath-Standard kennt nur einfache Teilstringsuche
 - contains()-Funktion
- Oracle-Erweiterung in XPath
 - ora:contains()
 - Bindet Oracle TEXT-Funktionalität ein
 - Einschränkungen beachten (XPath, Query Rewrite)

```
select rowid from NR_NACHRICHT_TAB e
where existsNode( value(e),
  '/nachricht/titel[ora:contains(text(),"Kooperation")>0] ',
  'xmlns:ora="http://xmlns.oracle.com/xdb" '
) = 1;
```

Oracle TEXT

Verhalten bei DML

- Indexsynchronisierung
 - Manuell (CTX_DDL.SYNC_INDEX)
 - Job-Gesteuert
 - On Commit (nicht zu empfehlen)
- Aber: "Transactional"-Index
 - In Memory-Suche der Pending-Dokumente
 - Abfrageergebnisse stets auf aktuellem Stand
 - Ermöglicht "Höheres" Synchronisierungsintervall





TRANSACTIONAL-Index

- Indexerstellung

```
create index idx_nachrichten_volltext
on nr_nachricht_tab (object_value)
indextype is CTXSYS.CONTEXT
parameters ('
  section group nachrichten_sec_group
  transactional
')
```

- Transactional-Query abschalten

```
begin
  ctx_query.disable_transactional_query := TRUE;
end;
```



Verwendung von MDATA

- MDATA-Sektionen, um Performanz von „Mixed Queries“ zu verbessern

```
SELECT id FROM idx_docs WHERE CONTAINS(text, 'Sommer  
AND MDATA(autor, (Nigella Lawson)')>0
```

-> werden nicht „tokenized“
-> sind transaktional veränderbar

- Abfrage nur mit Context Index verwendbar
- MDATA-Sections können nur in der XML_SECTION_GROUP verwendet werden.



Und dann war da noch ...

- XML DB: CTXXPATH-Index wird obsolet
- 11g kommt bald ...
 - Neuerungen bei TEXT
 - Neuerungen bei der XML DB
 - Compact Binary XML
 - Neuer Indextyp: "*XML Index*"



Weitere Informationen

- Oracle Dokumentation
 - Oracle TEXT Developers' Guide
 - Oracle TEXT Reference
 - Oracle XML DB Developer's Guide and Reference
- Oracle Technology Network (OTN)
 - Database → Content Management
- Metalink
 - Note 249991.1: Oracle TEXT technical Overview