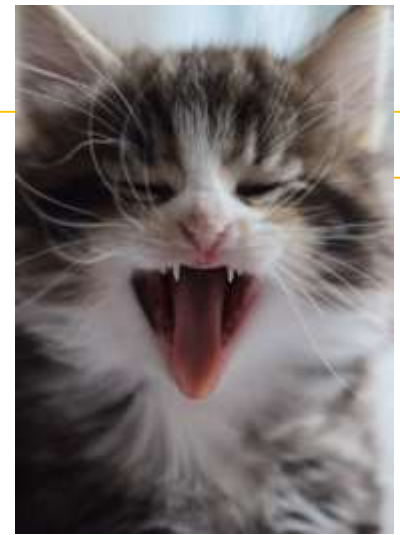


APEX connect



Oracle Text – 12c – Apex Integration – Binär Dokumente indizieren

**MIT ORACLE TEXT DOKUMENTE
INDIZIEREN UND IN APEX DARSTELLEN**

THESAURUS UND THEMES EINSETZEN

Gunther Pippèrr - IT-Architekt - Berater



Background

Gunther Pippèrr arbeitet seit mehr als 18 Jahre intensiv mit den Produkten der Firma Oracle im Bereich Datenbanken/Applikationsserver und Dokumenten-Management und seit 4 Jahren mit Oracle Primavera.

Herr Pippèrr hat sich tiefes Wissen über den Aufbau komplexer IT Architektur aneignen können und hat dieses in der Praxis erfolgreich umgesetzt.

Herr Pippèrr hat eine Abschluss als Dipl. Ing. Technische Informatik (FH) an der FH Weingarten.

Functional Expertise

- IT System Architekt
- Technische Projektleitung
- Design und Implementierung von Datenbank Anwendungen
- Entwurf und Umsetzung von IT Infrastrukturen zum Datenmanagement

Industry Expertise

- High-Tech
- Real Estate
- Utility
- Communications
- Pharm.

Selected Experience

- Administration und Wartung von Oracle Datenbank Umgebungen
- Datenbank Architekt für ein Projekt zur Massendatenverarbeitung in der Telekommunikation
- Betrieb und technische Projektverantwortung für IT Infrastrukturprojekte, z.B.:
 - Unterstützung beim Betrieb der Datenbank Umgebung für das größte deutsche Kunden Bindungsprogramm
 - Zentrale Datenhaltung für Münchner Hotelgruppe mit über 25 Hotels weltweit,
 - Redundante Cluster Datenbank Infrastrukturen für diverse größere Web Anwendungen wie Fondplattform und Versicherungsportale
- Architekt und technische Projektverantwortung für ein Smart Metering Portal für das Erfassen von Energiezählerdaten und Asset Management
- Architekt und Projektleitung , Datenbank Design und Umsetzung für die Auftragsverwaltung mit Steuerung von externen Mitarbeitern für den Sprachdienstleister von deutschen Technologiekonzern

Ich bin Ihr Partner für komplexe Oracle Projekte und unterstütze Sie gerne

Agenda

- 1 **Oracle Text Übersicht**
- 2 Die Aufgabenstellung
- 3 Daten laden und indizieren
- 4 Für die Schlagwortsuche einen Thesaurus und einen Theme Index verwenden und pflegen
- 5 Dokumente in Apex suchen und anzeigen



Historie:

- SQL*TextRetrieval™ 1.0, 1990
- SQL*TextRetrieval™ 2.1, 1993
- TextServer 3, 1994
- TextServer 3.1, 1995
- Oracle ConText Option 1.1, Oracle7 7.3.2.2, 1996
- Oracle ConText Option 2.0, Oracle7 7.3.3, 1997
- Oracle ConText Cartridge, Oracle8 8.0.3, 1997
- Oracle ConText Cartridge 2.3, Oracle7 7.3.4, Oracle8 8.0.4
- Oracle ConText Cartridge 2.4.5 Oracle 8.0.5
- Oracle interMedia Text8i Release 1 (8.1.5)
- Oracle interMedia Text8i Release 2 (8.1.6)
- Oracle Text (9.x,10g,11g,12c)

Wie einfach funktioniert Oracle Text?

Beispiel:

```
create table docs
  (id number primary key, text varchar2(80))
```

Primär
Schlüssel

Text
Spalte

```
insert into docs values ( 1 , 'Thomas May'); commit;

create index idx on docs(text) indextype is ctxsys.context;
```

```
select id from docs where contains (text, 'May')>0;
```

Unterstützte Datentypen:

VARCHAR2, LONG, LONG RAW, BLOB, CLOB, CFILE

Demo

Aufbau eines Volltext Index vom Typ CONTEXT

- Aufbau Context Index aus einzelnen Tabellen

DR\$<index_name>\$<?>

Oracle
Text
Domain
Index

Typ:
ctxsys.context

\$I => invertierte Liste aller Wörter/Tokenen

Index DR\$<index_name>**\$X auf \$I**

\$K => Key Map – rowid's zu internal docid's

\$R => „Reverse lookup“ Mapping-Tabelle

\$N => Gelöschte Datensätze vermerken

Optional

\$P => Substrings

Optional ab 12c

\$O => Forward Indexing für Highlighting, Snippet, Markup Function)

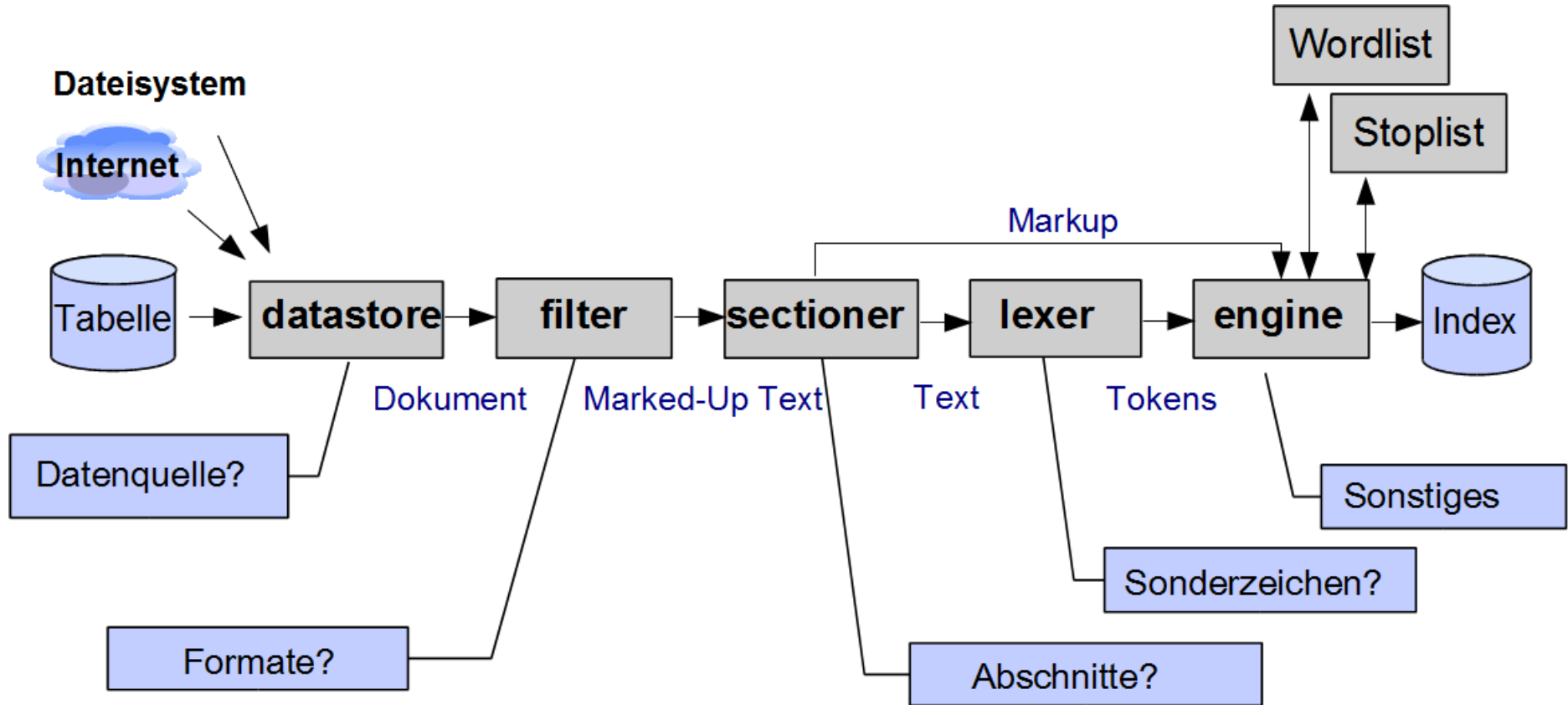
\$D => Forward Indexing , kompletter Text als PlainText

\$G => Near Real Time Indexes für schnelles Sync on Commit

Die Eigenschaften der Tabellen werden mit STORAGE Properties gesetzt

Der Pipeline-Prozess beim Anlegen

- Create index ... type `ctxsys.context` (.. Paramter ..)



Ablauf Insert/Update der Text-Tabelle

1. Insert oder Updates der Texttabelle DR\$user_index\$N

2. Wird eingestellt in Queue

3. User commit

4. Update der Index-Tabellen mit alter index oder im Batch-Modus

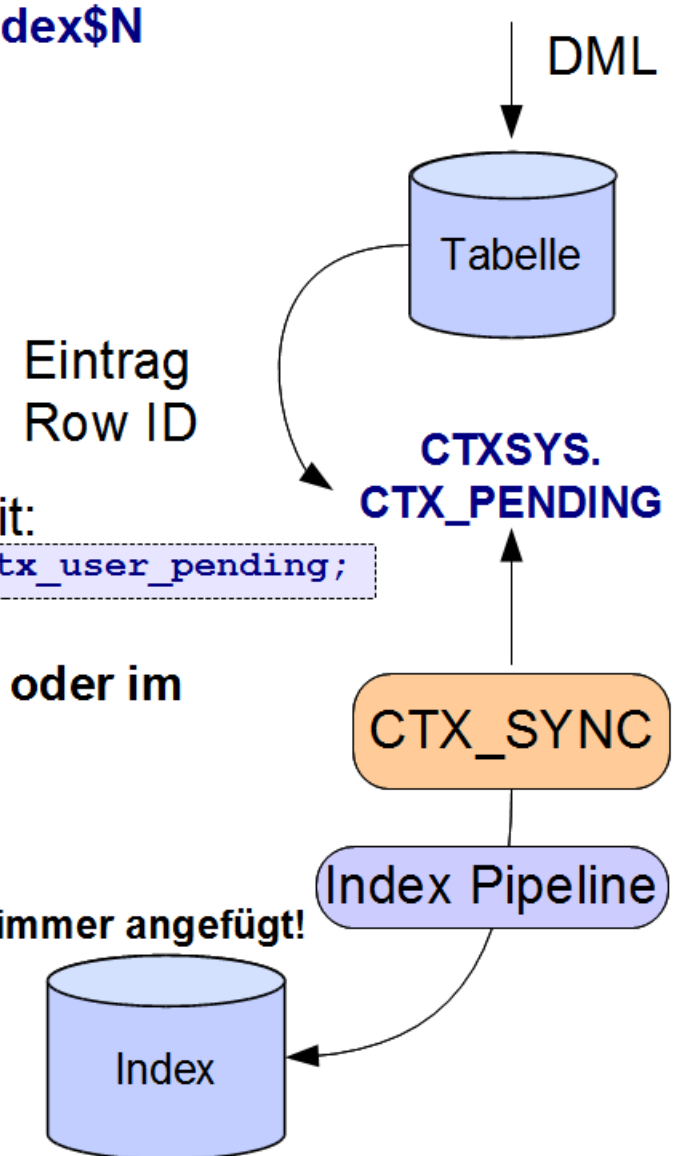
```
ctxsys>Exec CTX_DDL.SYNC_INDEX
```

```
alter index indname rebuild  
online  
parameters ('sync');
```

Überwachen mit:

```
select * from ctx_user_pending;
```

Token werden immer angefügt!



Suchen – Der Oracle Contains Operator

- Contains Operator Features



```
SELECT id
       , SCORE(1)
  FROM TEXTE
 WHERE CONTAINS(COLUMN, 'query_string',1)>0
/
```

-- Beispiel für die fussy Suchen in der ausführlichen Variante

```
SELECT * FROM texte WHERE contains(text, 'fuzzy(Hunt, 50, 50, WEIGHT)') >0;
```

Demo

Oracle 12c – Near Real Time Index (1)

■ Near Real Time Index

- Neue Tabelle \$G und Index \$H
- Index in zwei Teile aufgeteilt – Stage (möglichst in Memory) und dann später ein Merge in den eigentlichen Index

Preferences anlegen:

```
exec ctx_ddl.create_preference('GPI_STORAGE', 'BASIC_STORAGE')
exec ctx_ddl.set_attribute('GPI_STORAGE', 'STAGE_ITAB', 'true')
```

```
exec ctx_ddl.set_attribute('GPI_STORAGE', 'G_TABLE_CLAUSE', 'storage
(buffer_pool keep)')
exec ctx_ddl.set_attribute('GPI_STORAGE', 'G_INDEX_CLAUSE', 'storage
(buffer_pool keep)')
```

Index mit Sync on Commit anlegen:

```
create index idx_doc_files ON documents(FilePointer)
indextype is ctxsys.context parameters('storage GPI_STORAGE sync (on
commit)')
```

Oracle 12c – Near Real Time Index (2)

- Mergen bei Bedarf:

```
execute ctx_ddl.optimize_index(idx_name => 'idx_doc_files'  
                               , optlevel => 'MERGE')
```

- Job einrichten mit:

```
exec ctx_ddl.add_auto_optimize('idx_doc_files' )
```

Oracle 12c - Index Forwarding

- Index Forwarding – Snippets/ Highlight vom Text beschleunigen
 - Plain Text in \$D speichern
 - Forward Indexing in \$O ablegen

```
exec ctx_ddl.create_preference('GPI_STORAGE ', 'BASIC_STORAGE' )  
  
exec ctx_ddl.set_attribute      ('GPI_STORAGE ', 'FORWARD_INDEX', 'TRUE' )  
exec ctx_ddl.set_attribute      ('GPI_STORAGE ', 'SAVE_COPY', 'PLAINTEXT' )
```

Leider gleich mal auf eine Bug gelaufen
Cursor von `CTX_DOC.SNIPPET` werden nicht geschlossen – **BUG 20892798**
Kein FIX öffentlich – Falls gleicher Fehler auftritt erneuten Bug mit Prio eröffnen!

Oracle 12c - Pattern Stopclass

■ Pattern Stopclass

- Verwendung von Pattern/Regulären Ausdrücken für Stop Words
 - Alle Nummern größer als 4 Zeichen nicht indizieren

```
ctx_ddl.add_stopclass ('GPI_STOPLIST', 'NUM_GT_4', '[:digit:]\,\.\]{5,}') 
```

- Alle Wörter länger als 25 Zeichen

```
ctx_ddl.add_stopclass('GPI_STOPLIST', 'WORD_GT_25', '[:alnum:]]{25,}') )
```

Alternativ einfach alle Zahlen (bereits ab 11gR2)

```
ctx_ddl.add_stopclass('GPI_STOPLIST', 'NUMBERS');
```

Nur auf eigene STOP Word Listen möglich!

Unser Aufgaben Stellung für das Apex Archiv

- Verweise auf Dokument in die Datenbank laden
- Text aus dem Dokumente extrahieren und diesen Text indizieren
- Inhalte der Dokumente bewerten und Dokumente Schlagwörtern zuordnen
- In einer Apex Maske die Dokumente suchen und anzeigen
- In Apex dynamischen Baum (Tree Element) aus der Dokument Verschlagwortung erstellen



Wo die Dokumente speichern?

In der Datenbank

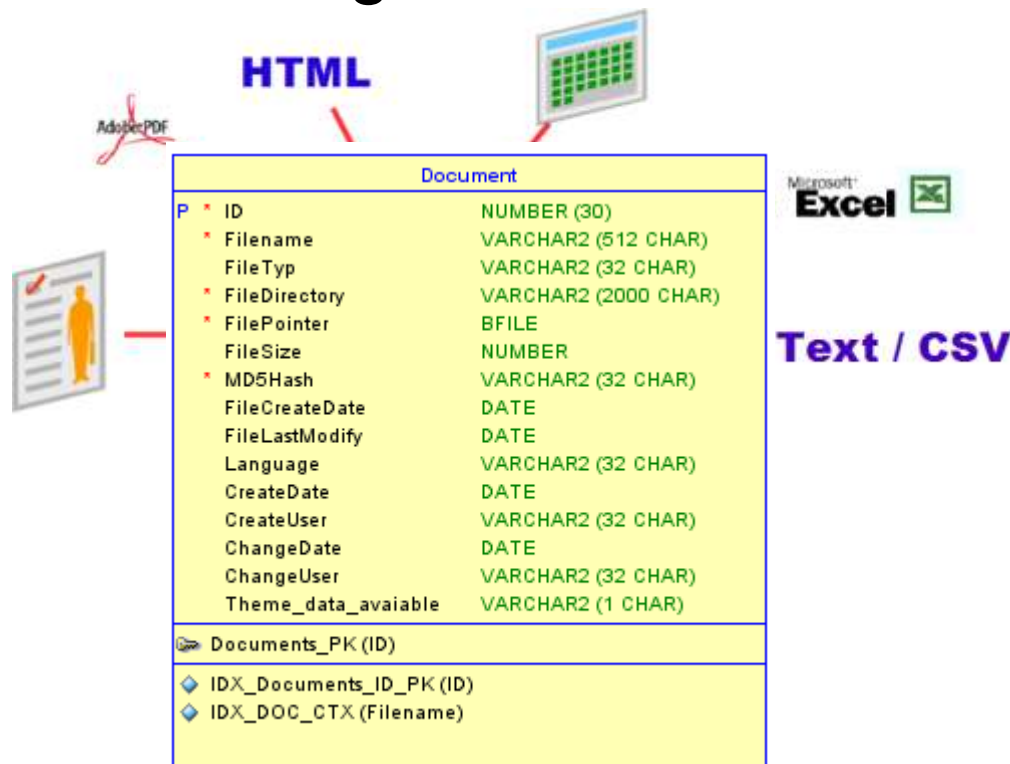
- **Vorteil:**
 - Sicherheit und Zugriffskontrolle an einer zentralen Stelle
- **Datentyp : Oracle Secure File (BLOB)**
 - Inkl. Deduplizierung und Komprimierung in der Datenbank

Im Storage

- **Vorteil:**
 - Können weiter extern bearbeitet werden
 - Die Datenbank bleibt klein
- **Datentyp: BFILE**
 - Nur ein Pointer auf eine Datei wird in der Datenbank gespeichert
 - Auf den Pointer wird der Index aufgebaut

Wie die Dokumente laden

- Beim Laden möglichst viele Eigenschaften der Dokument ermitteln und speichern.
- Zum Beispiel mit Hilfe von „Magic File“ Informationen und Python



Code : https://www.pipperr.de/dokuwiki/doku.php?id=prog:python_data_loader_oracle_bfile_text_index

Den Text aus den Dokument auslesen

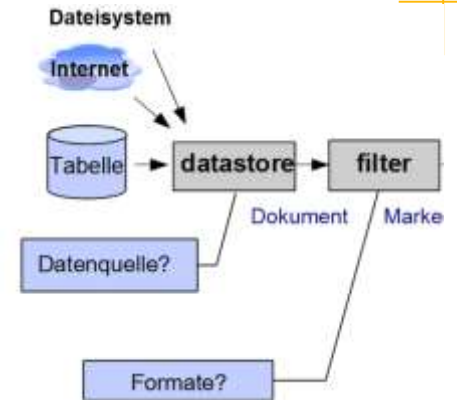
Was bedeutet Filtern?



Lorem ipsum dolor sit amet,
consectetur adipiscing elit Nullam
at ulla bibendum condimentum
sapien in congue diam.



Binäres Dokument → Filter → In den Index übernehmen



AUTO_FITER - Oracle Outside In Technology

■ AUTO_FITER

- In den Oracle Core als Library eingebunden

```
select owner,LIBRARY_NAME from dba_libraries where owner ='CTXSYS';  
OWNER          LIBRARY_NAME  
-----  
CTXSYS         DR$LIB
```

Test mit:

```
$env:ORACLE_HOME/bin/ctxhx.exe apex_connect2016.pptx output_data.html
```

Lizensierung als Stand Alone? Unklar

Einzel auch installierbar

=> <http://www.oracle.com/technetwork/middleware/webcenter/content/oit-dl-otn-097435.html>

USER_FILTER - Dokumente selber filtern

- Über eigene Skripte die Dokumente in Text zerlegen
- Möglichkeiten
 - Wrapper über die Oracle Outside In Technologie
 - Vorteil: Fehlerbehandlung dynamisch implementieren!
 - Eigene Software / Fremd Hersteller für spezielle Datentypen einbinden
 - Wie eine eigene PDF Library z.B.
 - Foxit PDF IFilter – Server => <https://www.foxitsoftware.com>
 - PDFLib => <http://www.pdflib.com/products/tet/>
 - Adobe PDF iFilter 64
 - Und viele weitere



USER_FILTER Beispiel:

Sicherheitsproblem!

Über den Filter können wir jetzt im OS lesen!

- Script für den Aufruf des Filter in `$ORACLE_HOME/ctx/bin` hinterlegen
- Einfachstes Beispiel:

Achtung! Wird unter dem OS User aufgerufen unter dem die Datenbank läuft!
Auf entsprechende Rechte diese Users achten!

```
echo PARAM 1: %1 >> d:\temp\ctx_test.log
```

```
echo PARAM 2: %2 >> d:\temp\ctx_test.log
```

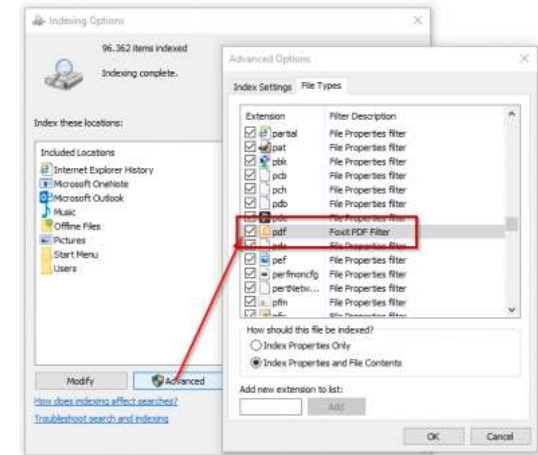
```
set ORACLE_HOME=D:\oracle\products\12.1.0.2\dbhome_1
```

```
%ORACLE_HOME%/bin/ctxhx.exe %1 %2 >> d:\temp\ctx_test.log
```

- Beim Aufruf wird **NUR** der Name des Dokuments (Keine Dateiendung!) und der erwartete Dateiname für den Spool out des Filter übergeben
- **Problem:** Keine weiteren Parameter möglich mit dem das Script gesteuert werden kann, wie Dateityp oder Codierung etc.

Ein Beispiel mit dem Microsoft IFilter

- Über das IFilter Interface (gedacht für den MS Indexer) können ddl's als Filter für Dokumentarten registriert werden
- Foxit PDF IFilter wird so für Oracle Text verfügbar
- Kann über FiltDump.exe (aus dem Windows SDK) können diese DDL's in Skripten angesprochen werden



Probleme mit "ORA-07445: exception encountered" schränken den Einsatz leider sehr ein!

Beispiel: Verschlüsseltes PDF führt sofort zu solch einem Fehler!

Probleme

- **ORA-07445: exception encountered: core dump [_intel_memcpy()+5376] [ACCESS_VIOLATION]**

Und schon ist man ratlos....

```
DROP INDEX idx_doc_files
*
ERROR at line 1:
ORA-29868: cannot issue DDL ON a DOMAIN INDEX marked AS LOADING

-- als sys anmelden

-- mark as failed
EXEC ctxsys.ctx_adm.mark_failed( 'GPI', 'IDX_DOC_FILES' );

DROP INDEX GPI.idx_doc_files;

INDEX dropped.
```

Fazit – wie die Dokumente filtern?

Mit Oracle Stellent

- Vorteil **AUTO_FILTER**:
 - Sehr vielen Dokumentformate
 - Voll integriert
 - Fehler führen nicht sofort zu einem **ORA-07445**
- **Nachteil**:
 - Intransparent
 - Teilweise unklar was hier wirklich geklappt hat und vor allem mit welchen Ergebnis

Mit eigenen Filtern

- Vorteil **USER_FILTER**:
 - Besondere Regeln umsetzbar
 - Behandlung von verschlüsselten Daten
 - PDF Spezial Fälle abdecken
- **Nachteil**:
 - Zusätzliche Software muss beschafft und gewartet werden
 - Nicht 100% stabil – Fehler schwer nachvollziehbar
 - Viele **ORA-07445 in ersten Tests – Nacharbeit mit dem Oracle Support notwendig**

Fazit – Verbesserungsvorschlag

- Auf Filtern aus der DB verzichten und Dokumente gleich beim Laden filtern
 - Dokument verbleibt auf Disk - BFILE Referenz
 - CLOB mit reinen Text Inhalt als HTML Dokument wird indiziert
 - Soll die DB klein bleiben, Text Dokument als BFILE
 - Vorschau erfolgt über diese CLOB Daten

Vorteil: Wandeln in Text kann so erfolgen, das sinnvoll und Nachvollziehbar auf die Fehler reagiert werden kann

Nachteil: Loader für die Dokumente wird komplexer

Oracle Thesaurus und Oracle Theme

Beschreibt die Relationen zwischen Wörtern

Thesaurus

`$ORACLE_HOME
ctx\data\enlx\drelsUS.dat`

Übersetzen

Knowledge Base

Oracle Text
LEXER

Verwenden

Basis für den
Theme Index

SQL
RT PL/SQL
Java
NT SCALA
NT Groovy

ABOUT(xxx) Query nun möglich

```
ctxload  
-user ctxsys/xxxxxx  
-thes  
-name T_PROD_DOCUMENT  
-file prog_lang.txt  
-thescase y
```

Alternativ PLSQL => CTX_THES

```
ctxkbtc  
-user ctxsys/xxxxxx  
-name T_PROD_DOCUMENT
```

```
CREATE INDEX  
idx_doc_files ON  
documents (FilePointer)  
INDEXTYPE IS  
CTXSYS.CONTEXT  
PARAMETERS ('LEXER  
GPI_LEXER')
```

Um was geht es in den Dokumente? - Theme

Zum Beispiel im Dokument 406

```
declare
  v_themes ctx_doc.theme_tab;
begin
  ctx_doc.themes(  INDEX_NAME => 'idx_doc_files',
                  , TEXTKEY    => '406'
                  , RESTAB=> v_themes
                  , FULL_THEMES => true , NUM_THEMES=>10);
  for i in 1..v_themes.count loop
    dbms_output.put_line(v_themes(i).theme||': '||v_themes(i).weight);
  end loop;
end;
/
```

```
:Linux:operating systems:computer software industry:computer industry:hard sciences:science and technology::49
:operating systems:computer software industry:computer industry:hard sciences:science and technology::11
:Oracle Corporation:computer software industry:computer industry:hard sciences:science and technology::39
:databases:information technology:computer industry:hard sciences:science and technology::6
:tests:consideration::16
:control:potential relations:abstract ideas and concepts::1
:command:control:potential relations:abstract ideas and concepts::15
:relation::1
:consideration::3
:change:dynamic relations:abstract ideas and concepts::12
```

Zusammenfassung - Gist

```
declare
  v_gklob clob;
  v_line varchar2(100);
  v_read number:=100;
begin
  ctx_doc.gist( index_name => 'IDX_DOC_FILES'
               ,textkey   => '405'
               ,restab    => v_gklob);

  dbms_lob.read(v_gklob, v_read , 1, v_line);
  dbms_output.put_line('-- : ' || trim(v_line));
  dbms_lob.freetemporary(v_gklob);
end;
/
```

Beispiel Ausgabe für das vorherige Dokument:

Oracle8i addresses this requirement by providing the industry's most comprehensive set of query

Verschlagwortung von Dokumenten

- Wie kann ich erkennen, dass es in einem Dokument über Oracle APEX Entwicklung geht?
 - Aus einem Thesaurus einen Theme Index als Regelwerk erstellen
 - Thesaurus
 - Beziehung zwischen Wörtern definieren wie „APEX“ = „Application Express“
- Schwierigkeit : Sehr dynamisch, wie pflegen und von wem?

Lösung: Apex Oberfläche für den Thesaurus

- Über ein Datenmodell wird der Thesaurus abgebildet
 - Über PL/SQL wird das dem Meta Informationen ein Thesaurus nach Bedarf jeweils neu aufgebaut

```
Thesaurus

▼ PROD_DOCUMENT
  Borland
  CoffeeScript
  Delphi
  Groovy
  ▼ Java
    -> Relation :NT to Groovy
    -> Relation :NT to SCALA

--- compile the thesaurus
---
$ ctxkbtc -user ctxsys/ctxsys -name T_PROD_DOCUMENT
---
```

```
ctx_thes.create_thesaurus (name      => trec.name
                          , casesens => false
                          );

--create a Phrase to the thesaurus
for prec in c_parse (p_ths_id => trec.id)
loop
  dbms_output.put_line ('-- Info :: create phrase ::' || prec.phrase);
  ctx_thes.create_phrase (tname      => trec.name
                        , phrase     => prec.phrase
                        );
end loop;

-- define a relation to this phrases
for rrec in c_relation (p_ths_id => trec.id)
loop
  ctx_thes.create_relation (tname      => trec.name
                          , phrase     => rrec.phrase
                          , rel        => rrec.relation
                          , relphrase  => rrec.rel_phrase
                          );
end loop;
```



Nachteil: Index muss neu aufgebaut werden, damit die Änderung auch im Theme Index sichtbar werden! Gesucht werden kann mit den neuen Werte trotzdem!

Herausforderung – Die Spracheinstellungen

- Beim Anlegen eines Thesaurus werden die Spracheinstellungen der Umgebung verwendet!

- Prüfen mit:

```
select sys_context ('USERENV', 'LANGUAGE') as NLS_LANG_Parameter  
from dual;
```

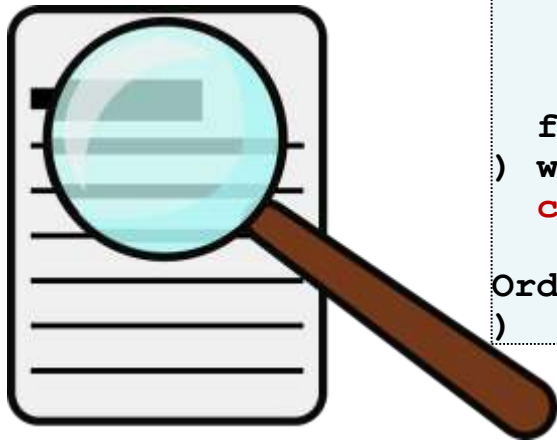
- Beim Anlegen des Theme Indexes werden ebenfalls die Spracheinstellungen berücksichtigt!
 - Ob es geklappt hat über die \$I Tabelle prüfen auf token_type 1

Fazit – Einsatz Thesaurus und Theme

- Vorteil:
 - Thesaurus als Basis für einen Dokumenten Such Baum
- Nachteil
 - Mehrsprachlichkeiten nicht so ganz einfach zu implementieren trotz MULTI_LEXER
 - Theme Index – Nachvollziehbarkeit, ob es immer so wie gewünscht klappt, schwierig

Dokumente mit Oracle Text in APEX finden

- Oracle Text in APEX einbinden
 - Einfach in der SQL Abfrage den CONTAINS Operator verwenden!



```
select *
  from (
select  FILETYP
        ,FILENAME
        ,FilePointer
        ,FILEDIRECTORY
        ,FILESIZE
        ,score(1) as ctx_sore
  from DOCUMENTS
) where (
  contains( FilePointer
            , nvl(:P10_REPORT_SEARCH, 'oracle',1 ) > 0
Order by ctx_score desc
)
```

Demo

Treffer im Dokument in der Trefferlist anzeigen

- Mit `CTX_DOC.SNIPPET` Teilergebnisse in der Treffermenge farblich markiert darstellen

```
EXEC ctx_doc.set_key_type('ROWID');  
  
-- Abfragen und Ergebniss mit ermitteln  
  
SELECT CTX_DOC.SNIPPET(  
    index_name => 'IDX_DOC_FILES'  
    ,textkey    => rowid  
    ,text_query => '?Hunt'  
    ,starttag  => '<span style="color:#034F84; background-  
color:#F7CAC9">'  
    ,endtag    => '</span>'  
    ,separator => '<b>...</b>') AS textfragment  
FROM texte WHERE contains(text, '?Hunt') >0  
/
```

Ab Oracle 10g

Und wieder Probleme:

- ORA-01000: maximum open cursors exceeded
 - Bug 20892798 : MANY CURSORS OPENED WHEN USING FORWARD INDEXING LEADING TO ORA-01000 ERROR
=> Product Version 12.1.0.2

```
at java.lang.Thread.run(Unknown Source)
Caused by: java.sql.SQLException: ORA-00604: error occurred at recursive SQL level 1
ORA-01000: maximum open cursors exceeded
ORA-00604: error occurred at recursive SQL level 1
ORA-01000: maximum open cursors exceeded
ORA-01000: maximum open cursors exceeded
```

Workaround: Forward Index abschalten und weiteren Bug bei Oracle eröffnen!

Dokument in HTML wandeln

- Funktioniert auch ohne Indizierung!
- Filter Präferenzen definieren

```
BEGIN
  -- create the policy for selffiltering
  ctx_ddl.create_preference(preference_name => 'fast_filter'
                           , object_name   => 'AUTO_FILTER');

  ctx_ddl.set_attribute(preference_name => 'fast_filter'
                       , attribute_name => 'OUTPUT_FORMATTING'
                       , attribute_value => 'FALSE');

  ctx_ddl.create_policy(policy_name      => 'GPI_FAST_POLICY'
                       , FILTER         => 'fast_filter');
END;
/

Begin
...
ctx_doc.policy_filter('GPI_FAST_POLICY', v_blob_doc, v_doc_text, FALSE);
...
End;
/
```

Vollständigen Code siehe : https://www.pipperr.de/dokuwiki/doku.php?id=prog:oracle_text_in_plsql

12c – Sprache eines Text Fragments erkennen

- Preferences richtig setzen ! – Sonst **ORA-07445**

```
begin
ctx_ddl.create_preference( preference_name => 'GPI_LANG_LEXER'
                          , object_name    => 'AUTO_LEXER' );
ctx_ddl.create_policy    ( policy_name    => 'GPI_LANG_POLICY',
                          , lexer        => 'GPI_LANG_LEXER' );
end;
/
```

```
-- Do now something with the pure TXT data of the document
CTX_DOC.POLICY_LANGUAGES (
  policy_name    => 'GPI_LANG_POLICY'
  , document     => substr(v_doc_text,1,32000)
  , restab       => v_ctx_restab
);

--_ctx_restab
for i in 1..v_ctx_restab.count loop
  dbms_output.put_line('Language: ' || rpad(v_ctx_restab(i).language,20)
                       || ' Score:'
                       || lpad(v_ctx_restab(i).score,3));
end loop;
```

Language: american	Score: 87
Language: romanian	Score: 86
Language: german	Score: 86
Language: dutch	Score: 86
Language: french	Score: 85

Apex Tree Element für einen Dokumenten Baum

- Apex Baum Element für die Suche nach Schlagwörtern verwenden

```
▼ T_PROD_DOCUMENT Search for :: oracle
  Borland
  CoffeeScript
  ▼ Delphi
    -> Found :linux1_nt1.pdf in Delphi
  Groovy
  ▼ Java
    -> Found :11g_new_features.pdf in Java
    -> Found :12coracletexttp-1961244.pdf in Java
    -> Found :2006_05_17_SQL_und_PL-SQL_SQL-Developer_und_mehr_Ulrike_Schwinn.pdf in Java
    -> Found :20120725-regio_muenchen-lange-ovm3.pdf in Java
    -> Found :20130723-regio_muenchen-thiem-oracle12c.pdf in Java
    -> Found :8i-816.ppt in Java
    -> Found :8ibackup.html in Java
    -> Found :FragenGesamt_dba1.pdf in Java
    -> Found :Integrigy_Oracle_Listener_TNS_Security.pdf in Java
```



SQL: siehe

https://www.pipperr.de/dokuwiki/doku.php?id=prog:oracle_apex_oracle_text_document_archive#tree_ansicht_ueber_die_eintraege_im_thesaurus

Fazit – Einsatz von Oracle Text



- Ein Top Feature der Oracle Datenbank
- Teilweise schwierig in der Problembehandlung
- Einfach in die APEX Welt integrierbar

I **Fragen** & A



**Binär Dokumente indizieren
und suchen**

Quellen

- Oracle Dokumentation und Support Portal
- https://www.pipperr.de/dokuwiki/doku.php?id=prog:oracle_apex_oracle_text_document_archive



- Wieder mal eine andere Script Library
 - <https://orapowershell.codeplex.com/>



- Bildmaterial : <https://pixabay.com>

Question & Answer

Discussion Area in Jazz 2

*You don't have to put an abrupt stop to exciting discussions at the end of a talk:
In Room "Jazz 2" you can keep conversation going.*

