

Oracle als Baukasten aus Standardsoftware für Individuallösungen

Vorstellung



Gerhild Aselmeyer
Dipl. Math.

Seit über 15 Jahren berate und unterstütze ich Unternehmen bei der Umsetzung von EDV-Projekten. Vor allem, wenn Sie historisch gewachsene Dateninseln verbinden möchten, um Ihre Geschäftsprozesse effizienter zu gestalten, biete ich Ihnen:

- **A**nalysen Ihrer Softwarelandschaft (Datenfluss, Performance, Vermeidung von Mehrfacheingaben)
- **K**onzepte zur Verknüpfung (Schnittstellen zur Datenintegration, EAI und zukünftig auch SOA)

& Unterstützung bei der

- **A**nwendungsentwicklung (inklusive Projektleitung)

mit Schwerpunkten bei

- Datenhaltung in SQL-Datenbanken (z.B. Oracle),
- Technischen Anwendungen (z.B. Fahrzeugverwaltung, Interprozesskommunikation)

4 Fragen zur Analyse der IT-Unterstützung von Geschäftsprozessen

Partnerschaften & Netzwerke

Mitglied im BeraterNet e.V.

Wissen. Austausch. Erfolg.

Die Oracle-Konferenz

DOAG 2012 Konferenz + Ausstellung
20. - 22. November 2012
Nürnberg Convention Center NCC Ost

400 Fachvorträge für DBAs, Developer und Architekten
Top-Keynotespeaker

Ich bin als Referent dabei

Agenda

1. Datenbankaufbau - Tabellen & Views

2. Verarbeitungslogik - inklusive der
Schlagwortindizierung

3. Die Anwenderschnittstelle – APEX

jeweils

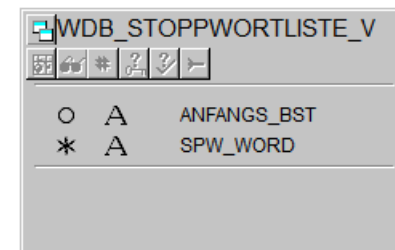
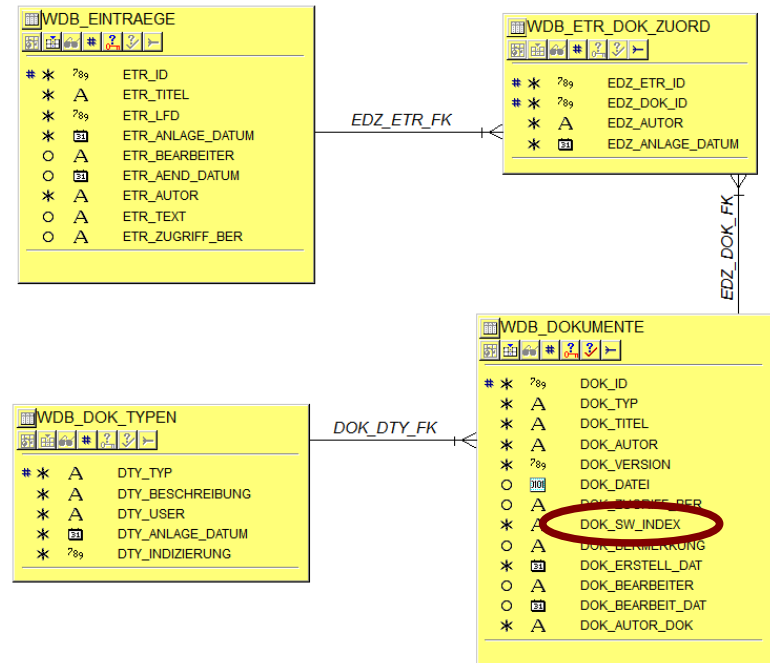
- für Textrecherche mit Oracle Text
- zum Arbeiten mit einem externen Abfrageergebnis im XML-Format
- Zugriffskontrolle mit VPD

Aufbau der Datenbank

- Wenige Tabellen
- Bis auf eine Spalte einfache Standarddatentypen
 - Ausnahme: XMLTYPE,
eine von Oracle mitgelieferte TYPE-Definition für XML-Dokumente
- Views für verschiedene Zugriffe auf Daten mehrerer Tabellen oder auf eine Oracle-Tabelle

Datenhaltung Textrecherche

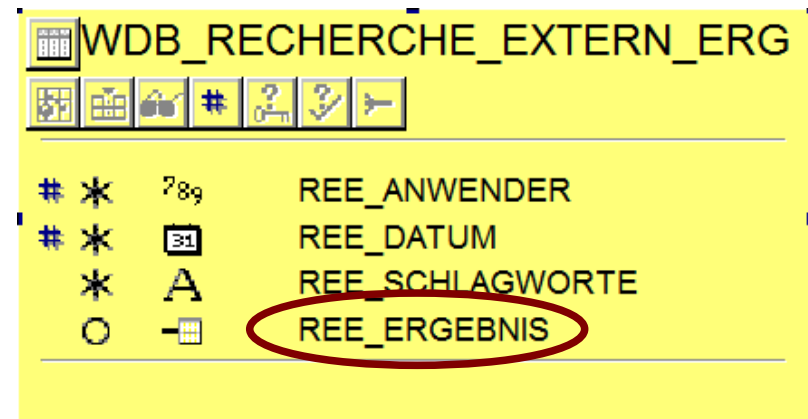
- Keine besonderen Datentypen
 - außer BFILE oder BLOB für Dokumente
- DOK_SW_INDEX zum Kennzeichen, ob und wie Dokument zu indizieren ist
- View für Stoppworte basiert auf Oracle Text Tabelle CTX_USER_STOPWORDS



WDB_STOPPWORTLISTE_V		
○	A	ANFANGS_BST
*	A	SPW_WORD

Datenhaltung externe Recherche


- Kleine Tabelle
- Speicherung des Abfrageergebnis
 - komplett in
 - REE_ERGEBNIS
 - definiert als XMLTYPE



WDB_RECHERCHE_EXTERN_ERG			
# *	?	89	REE_ANWENDER
# *	31		REE_DATUM
*	A		REE_SCHLAGWORTE
O	-		REE_ERGEBNIS

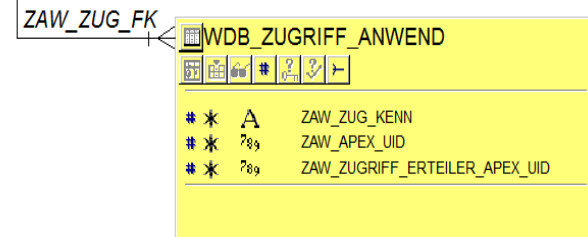
Datenhaltung Zugriffskontrolle

- WDB_ZUGRIFFE enthält Listen mit Kennzeichen für
 - *Gobal*
 - *Persönlich*
- WDB_ZUGRIFF_ANWEND enthält die Zuordnung von Nutzern zu Listen
- Die Views bilden verschiedene Zugriffsszenarien ab

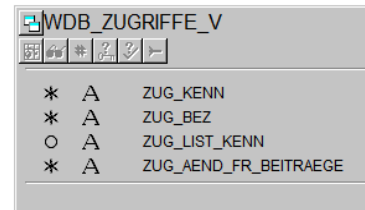


WDB_ZUGRIFFE	
# *	A ZUG_KENN
*	A ZUG_BEZ
o	A ZUG_LIST_KENN
*	A ZUG_AEND_FR_BEITRAEGE
o	A ZUG_BERMERKUNG

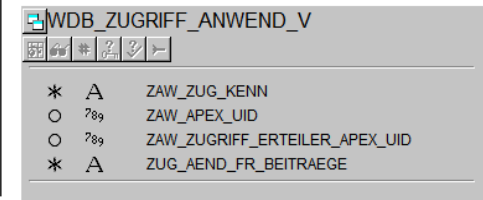
ZAW_ZUG_FK



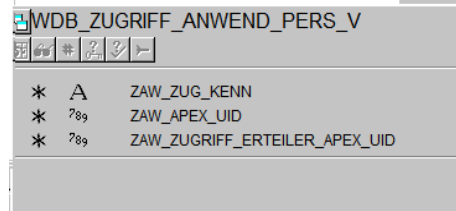
WDB_ZUGRIFF_ANWEND	
# *	A ZAW_ZUG_KENN
# *	?% ZAW_APEX_UID
# *	?% ZAW_ZUGRIFF_ERTEILER_APEX_UID



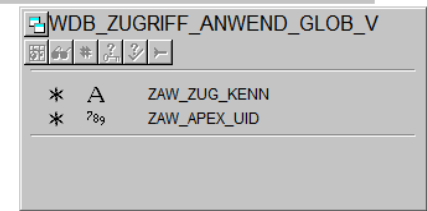
WDB_ZUGRIFFE_V	
*	A ZUG_KENN
*	A ZUG_BEZ
o	A ZUG_LIST_KENN
*	A ZUG_AEND_FR_BEITRAEGE



WDB_ZUGRIFF_ANWEND_V	
*	A ZAW_ZUG_KENN
o	?% ZAW_APEX_UID
o	?% ZAW_ZUGRIFF_ERTEILER_APEX_UID
*	A ZUG_AEND_FR_BEITRAEGE




WDB_ZUGRIFF_ANWEND_PERS_V	
*	A ZAW_ZUG_KENN
*	?% ZAW_APEX_UID
*	?% ZAW_ZUGRIFF_ERTEILER_APEX_UID



WDB_ZUGRIFF_ANWEND_GLOB_V	
*	A ZAW_ZUG_KENN
*	?% ZAW_APEX_UID

Performance & Design

- Anwendungsperformance beginnt mit dem Design
 - Am langsamsten ist die *'Physik'*
- 
- Natürlicher Schlüssel in jeder Tabelle
 - Tabellen und Indices in verschiedenen Tablespaces
 - Tabellen mit sehr unterschiedlichen Satzlängen in verschiedenen Tablespaces
 - Indexorganisierte Tabellen zu Indices
 - Storage-Parameter gezielt einsetzen (s. Oracle Doku)

Verarbeitungslogik

- Verschlagwortung mit Oracle Text
 - Vorbereitung
 - Indizieren
 - Weitere Programmierarbeiten
- Externe Recherche
 - Vorbereitung
 - Die Abfrage (FUNCTION getXMLSpec)
 - XML-Dokument verarbeiten
- Zugriffskontrolle mit VPD
 - Die Richtlinienfunktion
 - Berücksichtigung Neuerstellung Schlagwortindex
 - Restriktionen einrichten

Verschlagwortung mit Text

1. Vorbereitung

- Dem DB-Schema die notwendigen Rechte erteilen
- Erstellung von Vorlagen für eigene Präferenzen

```
BEGIN
```

```
    CTX_DDL.CREATE_PREFERENCE('Grundeinstellung_LEX',  
                              'BASIC_LEXER');  
    CTX_DDL.CREATE_PREFERENCE('Grundeinstellung_SWL',  
                              'BASIC_WORDLIST');  
  
    CTX_DDL.CREATE_STOPLIST('Grundeinstellung_SWS');  
  
    CTX_DDL.CREATE_PREFERENCE('Grundeinstellung_SPE',  
                              'BASIC_STORAGE');  
  
    CTX_DDL.SET_ATTRIBUTE('Grundeinstellung_SPE',  
                          'I_INDEX_CLAUSE',  
                          'tablespace WDB_INDEX_01 compress 2');  
    ...
```

```
END;
```

Verschlagwortung mit Text

2. Indizieren

```
CREATE INDEX DOK_SW_IDX1  
ON WDB_DOKUMENTE(DOK_DATEI)  
INDEXTYPE IS CTXSYS.CONTEXT ONLINE  
PARAMETERS('lexer Grundeinstellung_LEX wordlist Grundeinstellung_SWL  
storage Grundeinstellung_SPE stoplist Grundeinstellung_SWS FORMAT  
COLUMN DOK_SW_INDEX');
```

- Das Dokumentenformat ist
 - TEXT für HTML oder einfachen Text,
 - BINARY für alle anderen für Indizierung unterstützten Dokumente (PDF, MS DOC, RTF u.v.m),
 - IGNORE für nicht unterstützte Formate wie Bilder.

Neuaufbau Indices

```
procedure NeuaufbauIndices(sListenname varchar2, sIndexliste varchar2) is
  aIndexliste APEX_APPLICATION_GLOBAL.VC_ARR2 :=
  APEX_UTIL.STRING_TO_TABLE(sIndexliste, ',');
  WDB_Schema      all_tables.owner%type;
  RT_Schema       all_tables.owner%type;
  sAlterStatement varchar2(255);
begin
  if ( nvl(v('APP_USER'), user) <> WDB_Schema ) then
    SELECT SYS_CONTEXT('USERENV', 'CURRENT_SCHEMA')
      into RT_Schema FROM DUAL;
    EXECUTE IMMEDIATE 'alter session set current_schema='|| WDB_Schema;
  end if;
  for i in 1..aIndexliste.count loop
    sAlterStatement
      := 'ALTER INDEX '|| aIndexliste(i) ||
        ' REBUILD ONLINE PARAMETERS ('' '|| 'REPLACE STOPLIST '||
        upper(sListenname) ||'') PARALLEL 3';
    execute immediate sAlterStatement;
  end loop;
  if ( nvl(v('APP_USER'), user) <> WDB_Schema ) then
    EXECUTE IMMEDIATE 'alter session set current_schema='|| RT_Schema;
  end if;
end;
```

Weitere Programmierarbeiten

Textrecherche ...

- sind notwendig wenn man
 - APEX-Pakete unabhängig von der Oberfläche einsetzt
 - WORKSPACE anlegen
 - PROCEDURE APEX_Ini
 - Dokumente als BFILE speichert
 - FUNCTION DateiTransfer

Leider kann UTL_FILE auch in 11g noch immer nur VARCHAR2 speichern – also muss zerlegt werden

Externe Recherche

1. Vorbereitung

- ACL für Wissensdatenbank (WDB)
 - durchführen als SYS oder einem Schema mit DBA-Rolle
 - anlegen mit
DBMS_NETWORK_ACL_ADMIN.CREATE_ACL
 - zuweisen mit
DBMS_NETWORK_ACL_ADMIN.ASSIGN_ACL

FUNCTION getXMLSpec

/* externe Abfrage zusammenstellen */

```
sRequest          varchar(255)
:= 'http://services.dnb.de/sru/accessToken~' ||
  sZugangscode ||
  '/dnb?version=1.1' ||
  chr(38) |
  'operation=searchRetrieve' ||
  chr(38) |
  'query=' || replace(sSchlagworte, ' ', '+');
```

FUNCTION getXMLSpec

/* Ergebnis der externen Abfrage in 2000-
Byte Stücken abholen und speichern */

```
tXMLFilePiceTable :=  
  UTL_HTTP.REQUEST_PIECES(sRequest, 100,  
                           sProxyFull);  
  if ( tXMLFilePiceTable.count > 0 ) then  
    for i in 1..tXMLFilePiceTable.count  
    loop  
      cXMLOut := cXMLOut ||  
                tXMLFilePiceTable(i);  
    end loop;  
  end if;
```


XML-Dokumente verarbeiten (1)

```
<!DOCTYPE ICECAT-interface SYSTEM
```

```
"http://data.icecat.biz/dtd/ICECAT-interface_response.dtd" >
```

- URI unter DOCTYPE-Anweisung = direkter Internetzugriff
- Keine Unterstützung für DTD in XMLDB
- daher lokale Speicherung nicht möglich.

→ Aus Sicherheitsgründen verweigert Oracle Speicherung des Dokumentes als XMLTYPE!

→ Lösung: Auskommentieren (im RETURN von getXMLSpec)

```
return replace(replace(replace(cXMLOut, sRequest, "),
```

```
'<!DOCTYPE',
```

```
'<!--DOCTYPE'), '.dtd">', '.dtd" -->');
```

XML-Dokument verarbeiten (2)

```
<?xml version="1.0" encoding="WINDOWS-1252"?>
<searchRetrieveResponse
  xmlns="http://www.loc.gov/zing/srw/">
  <version>1.1</version>
  <numberOfRecords>1883</numberOfRecords>
  <resultSetId>13479026520487585</resultSetId>
  <records>
    <record>
      ...
      <recordData> ...
        <dc:title>...</dc:title>
      ...
    </recordData> ...
  </record> ...
```

! enthält für den Oracle XMLParser keine Elementknoten

XML-Dokument verarbeiten (3)

! Jedenfalls liefert die Existenzabfrage in

```
function getXMLKnotenWert (nAnwenderID number, dDatum date,
                           sKnotenName varchar2, cXMLDoc clob)
    return varchar2 is ...
begin ...
    /* wenn ein XPath-Ausdruck in sKnotenname übergeben wurde
    * und der entsprechende Knoten existiert, dann ... */
    if ( sKnotenName is not null and
        tXMLNode.existsNode('//' || sKnotenName)>0 ) then
        ...
    end if;
    return sValue;
end;
```

! **0** für '*numberOfRecords*' oder '*recordData*' als sKnotenname

XML-Dokument verarbeiten (4)

→ Lösung:

1. '/'|| aus Existenzabfrage und anderen Zugriffen entfernen
2. XPATH-Ausdrücke wie folgt für sKnotenname verwenden:
 - 'node()/node()[2]/text()'
für '*numberOfRecords*'
 - '/node()/node()[4]/child::*[1]/node()[3]/child::*[1]/node()[1]'
für '*dc:title*'
 - '/node()/node()[4]/child::*[1]:P21_TITEL_LISTE ||
'/node()[3]/child::*[1]/node()'
für '*recordData*' mit Index für ausgewählten Titel

XML-Dokument verarbeiten (5)

! Soll der Return-Wert als Liste in APEX verwendet werden, bedenken Sie: das Standardtrennzeichen ':' kommt in XML-Knotennamen ebenso wie in Texten häufiger vor

! Verwenden Sie ein anderes:

```
for i in 1..nRecordAnz loop
    tXMLNode := DBMS_XMLDOM.ITEM(tXMLListe, i-1);
    sRecords := sRecords ||
                '# ' ||
                DBMS_XMLDOM.GETNODEVALUE (
                DBMS_XMLDOM.GETFIRSTCHILD (tXMLNode) );
end loop;
```

Zugriffskontrolle mit VPD – Die Richtlinienfunktion

```
function WDBInfo_Policy(obj_schema varchar2, obj_name varchar2)
return varchar2
is
    sPolicyClause varchar2(2000) := '';
...
begin ...
    sPolicyClause :=
        '(' || sColName_Pref || '_AUTOR = '' || sAPPUser || '' or ' ||
        sColName_Pref || '_ZUGRIFF_BER is null or ' ||
        sColName_Pref || '_ZUGRIFF_BER in ' ||
        '(select ZAW_ZUG_KENN from WDB_ZUGRIFF_ANWEND ' || ...;
...
return sPolicyClause;
```

Berücksichtigung Neuerstellung Schlagwortindex

```
function WDBInfo_Policy(obj_schema varchar2, obj_name varchar2)
return varchar2
is ...
begin ...
    if ( upper(sAPPUser) not in DBMS_APPLICATION_INFO.READ_MODULE (
        (upper(obj_schema), 'SYS', 'ADMIN') ) then
        ...
        sPolicyClause := '(' || sColName_Pref || '_AUTOR = '' || ...;
        ....
end;
procedure NeuaufbauIndices(sListenname varchar2,
                           sIndexliste varchar2)
...    if ( nvl(v('APP_USER'), user) <> WDB_Schema ) then ...
        DBMS_APPLICATION_INFO.SET_MODULE ('...', 'NbIndices');
```

Restriktionen einrichten

```
dbms_ols.add_policy
( object_schema    => '&&WDB_Schema',
  object_name      => 'WDB_DOKUMENTE',
  policy_name      => 'WDB_InfoPolicy',
  function_schema  => '&&WDB_Schema',
  policy_function  => 'WDB_Administration_pkg.WDBInfo_Policy',
  statement_types  => 'select',
  update_check     => FALSE,
  enable           => TRUE );
```


Zugriffskontrolle mit VPD oder Views

- Vorteile VPD
 - Zentrale Pflege in wenigen Funktionen statt in jeder View.
 - Bedingung flexibler, da zur Laufzeit als Text generiert.
 - Schutz der Daten bei GRANTs auf Tabellenebene an andere DB-Schemas.
 - Spätere Ergänzung ist einfacher, da
 - keine bzw. kaum Änderung im existierenden GUI (z.B. APEX-Seiten).
- Nachteile VPD
 - Nur in Enterprise Edition verfügbar.
 - Für das Einrichten von Richtlinien SYSDBA-Rechte notwendig

Die Anwenderschnittstelle - APEX

- Besonderheiten für das Einbinden von anderen *'Bausteinen'* aus der Datenbank
 - Stoppwortliste bearbeiten
 - Dokumente laden
 - Schlagwortrecherche
- Demo der Anwendung

Stoppwortliste bearbeiten


Page Name: [Stoppwortliste](#) Template: [Application default](#)
Title: [Stoppwortliste](#) Header Text:
HTML Header: Footer Text:
HTML Body: Build Option:
Help Text: Authorization: [No](#)
Page Group: Cached: [No](#)

Regions

Display Point: Page Template Body (1) ▾ ▲

10  [Administration](#) [List](#)

Display Point: Page Template Body (3)

10  [Stoppwortliste](#) [Report](#)

30  [Stoppwort ergänzen](#) HTML

Display Point: Region Position 01

1  [Breadcrumb](#) [Breadcrumb Entry](#)

Validations

Processes

After Submit

10	Stoppwort ergänzen	PL/SQL anonymous block	Conditional
20	Index anpassen	PL/SQL anonymous block	Conditional
30	Stoppworte löschen	PL/SQL anonymous block	Conditional

Branches

After Processing

10 [Go To Page 42](#) Conditional

- Report-Region Stoppwortliste mit Item für Auswahl des Anfangsbuchstaben
- 3 Prozesse, die gespeicherte Package-Module aufrufen

Dokumente laden

<h3>Page</h3> <p>Page Name: <u>Dokumente laden</u> Template: <u>Application default</u></p> <p>Title: <u>Dokumente laden</u> Header Text:</p> <p>HTML Header: Footer Text:</p> <p>HTML Body: Build Option:</p> <p>Help Text: <u>No help is available for this</u> Authorization: <u>No</u></p> <p>Page Group: Cached: <u>No</u></p>	<h3>Computations</h3> <p>After Submit:</p> <table><tr><td>10</td><td><u>P31_DOK_SW_INDEX</u></td><td>Unconditional</td></tr><tr><td>20</td><td><u>P31_DOK_VERSION</u></td><td>Conditional</td></tr><tr><td>30</td><td><u>F_AKTION</u></td><td>Unconditional</td></tr></table>	10	<u>P31_DOK_SW_INDEX</u>	Unconditional	20	<u>P31_DOK_VERSION</u>	Conditional	30	<u>F_AKTION</u>	Unconditional							
10	<u>P31_DOK_SW_INDEX</u>	Unconditional															
20	<u>P31_DOK_VERSION</u>	Conditional															
30	<u>F_AKTION</u>	Unconditional															
<h3>Regions</h3> <p>Display Point: Page Template Body (1) 10 <input type="checkbox"/> <u>Daten erfassen</u> <u>List</u></p> <p>Display Point: Page Template Body (3) 20 <input type="checkbox"/> <u>Dokument</u> HTML</p> <p>Display Point: Region Position 01 1 <input type="checkbox"/> <u>Breadcrumb</u> <u>Breadcrumb Entry</u></p>	<h3>Validations</h3> <table><tr><td>10</td><td><u>P31_DOK_TYP</u></td><td>Conditional</td></tr><tr><td>20</td><td><u>P31_DOK_TITEL</u></td><td>Conditional</td></tr><tr><td>30</td><td><u>P31_DOK_AUTOR_DOK</u></td><td>Conditional</td></tr></table>	10	<u>P31_DOK_TYP</u>	Conditional	20	<u>P31_DOK_TITEL</u>	Conditional	30	<u>P31_DOK_AUTOR_DOK</u>	Conditional							
10	<u>P31_DOK_TYP</u>	Conditional															
20	<u>P31_DOK_TITEL</u>	Conditional															
30	<u>P31_DOK_AUTOR_DOK</u>	Conditional															
	<h3>Processes</h3> <p>After Submit</p> <table><tr><td>10</td><td>Get PK</td><td>PL/SQL anonymous block</td><td>Conditional</td></tr><tr><td>30</td><td>Process Row of <u>WDB_DOKUMENTE</u></td><td>Automatic Row Processing (DML)</td><td>Conditional</td></tr><tr><td>35</td><td><u>DOK_DATEI</u> <u>ergänzen</u></td><td>PL/SQL anonymous block</td><td>Conditional</td></tr><tr><td>40</td><td><u>Syc SW-Index</u></td><td>PL/SQL anonymous block</td><td>Conditional</td></tr></table>	10	Get PK	PL/SQL anonymous block	Conditional	30	Process Row of <u>WDB_DOKUMENTE</u>	Automatic Row Processing (DML)	Conditional	35	<u>DOK_DATEI</u> <u>ergänzen</u>	PL/SQL anonymous block	Conditional	40	<u>Syc SW-Index</u>	PL/SQL anonymous block	Conditional
10	Get PK	PL/SQL anonymous block	Conditional														
30	Process Row of <u>WDB_DOKUMENTE</u>	Automatic Row Processing (DML)	Conditional														
35	<u>DOK_DATEI</u> <u>ergänzen</u>	PL/SQL anonymous block	Conditional														
40	<u>Syc SW-Index</u>	PL/SQL anonymous block	Conditional														

- Zusätzlicher Prozess, wenn Dokumente als BFILE gespeichert werden.

Schlagwortrecherche

Home > Application Builder > Application 117 > Page 2 > Edit Region

Region Definition | Report Attributes | Saved Reports | Print Attributes

Region: 3 of 4 Name: Recherche

Show All | Identification | User Interface | **Source** | Attributes | Header and Footer | Conditions | Security

Source

```
Region Source
SELECT SCORE(1)+SCORE(2) "Relevanz", DOK_TYP, DOK_TITEL, DOK_AUTOR,
       DOK_ERSTELL_DAT BEZ_DAT,
       BFile_Name(DOK_DATEI) Datei, DOK_VERSION,
       DOK ID "Link-ID", 31 "Seite", 'DOK' "Praefix"
FROM WDB_BERKUMENTE
WHERE CONTAINS(DOK_DATEI, :P2_SCHLAGWORT, 1) > 0
      OR CONTAINS(DOK_BERMERKUNG, :P2_SCHLAGWORT, 2) > 0
union
SELECT SCORE(1) "Relevanz", 'Text', ETR_TITEL, ETR_AUTOR,
       nvl(ETR_AEND_DATUM, ETR_ANLAGE_DATUM) BEZ_DAT,
       NULL Datei, ETR_LFD, ETR_ID "Link-ID", 32 "Seite", 'ETR' "Praefix"
FROM WDB_EINTRAEGE
WHERE CONTAINS(ETR_TEXT, :P2_SCHLAGWORT, 1) > 0
```

- SCORE-Parameter korrespondiert mit letztem Parameter von CONTAINS
=Trefferanzahl?
Nicht 100%ig!
- CONTAINS ist der Query-Operator für Indices des Typs CTXSYS.CONTEXT

Demo

- APEX live

Aus Erfahrung lernen – Vorhandenes nutzen

„Mathematiker sind faul – intelligent faul.“

Bevor sie mit der Lösung einer Aufgabe beginnen, verwenden sie zunächst lieber einige Zeit sich damit zu beschäftigen, was andere bereits zur (Teil-)Lösung erarbeitet haben. Kein Mathematiker erfindet das Rad ein zweites Mal.“

Vielen Dank für Ihre Aufmerksamkeit

F

Fragen

&

Antworten

A