

# Oracle Text 실습

## Document Control

날짜	작성자	Version	Change Reference
2014-8-20	고운용	1.0	최초 생성

## 순서

1. 실습 환경 준비
2. Context Application (Context Full-Text 검색)
3. Catalog Application (Catalog 검색)
4. Calssfication Application (주제 분류 검색)
5. Document Service

## 1. 환경 준비

### 1.1 Oracle VM Virtual Box 환경

- Oracle Virtual Box는 [오라클 홈페이지](#)에서 다운로드 가능하며, VM 이미지 파일은 별도로 제공합니다. 아래 환경정보는 제공되는 Oracle Virtual Box 이미지를 기준으로 작성된 환경정보입니다. Oracle VM Virtual Box가 아닌 환경인 경우에는 Oracle 12c 버전의 DB만으로도 진행이 가능합니다.

구분	내 용
OS version	Oracle Linux
Host IP	192.168.0.100
OS 계정/패스워드	oracle/welcome1
DB 버전	Oracle EE 12.1.0
DB SID	orcl
DB 계정/패스워드	myuser/welcome1

### 1.2 Oracle Text 설치

#### 1.2.1 CTX Package 설치

Oracle Text는 Oracle의 모든 Edition에서 사용이 가능합니다. 이 실습환경에서는 Oracle 12.1.0 버전을 사용하여 진행합니다. Oracle 12c에서는 CTX Package에 대한 별도 설치 작업없이 Oracle text를 사용할 수 있습니다. Oracle

11g 이하에서 Text 를 사용하기 위해서 수동으로 설치할 경우 다음과 같은 절차로 수행합니다.

- 우선, SQL\*Plus에 SYSDBA로 연결 한 후, 'catctx.sql'을 아래와 같이 실행하면 CTXSYS 스키마에 Text Dictionary가 생성됩니다.
- 이 작업이 끝나면 언어별로 적절한 default preference를 생성합니다. Oracle text가 지원하는 언어별 default preference는 \$ORACLE\_HOME/ctx/admin/defaults 디렉토리에 위치합니다. 이들 스크립트는 drdefXX.sql과 같은 파일 명을 갖는데, 여기서 XX는 사용하고자 하는 국제 코드입니다.
  - 예를 들어 한국의 경우엔 KR default preference를 수동으로 설치하기 위해서는 sqlplus에 CTXSYS 계정 으로 로그인 한 후, 'drdefkr.sql'을 실행시킵니다.

```
conn / as sysdba
spool text_install.txt
@?/ctx/admin/catctx.sql CTXSYS SYSAUX TEMP LOCK

connect CTXSYS/welcome1
@?/ctx/admin/defaults/drdefko.sql
spool off
```

### 1.2.2 CTX Package 설치 확인

Oracle Text 설치 여부는 다음과 같이 확인 합니다. 이 예제는 12.1.0 버전의 예제입니다.

```
conn / as sysdba;
column comp_name format a30;

select comp_name, status, substr(version,1,10) as version
from dba_registry where comp_id = 'CONTEXT';
COMP_NAME                                STATUS                                VERSION
-----
Oracle Text                             VALID                                12.1.0.2.0

select * from ctxsys.ctx_version;
VER_DICT                                VER_CODE
-----
12.1.0.2.0                             12.1.0.2.0

select substr(ctxsys.dri_version,1,10) VER_CODE from dual;
VER_CODE
-----
12.1.0.2.0

select count(*) from dba_objects where owner='CTXSYS';
COUNT(*)
-----
409

select object_type, count(*) from dba_objects
where owner='CTXSYS' group by object_type;
OBJECT_TYPE                                COUNT(*)
-----
SEQUENCE                                    3
PROCEDURE                                   2
OPERATOR                                    6
```

LOB	4
LIBRARY	1
PACKAGE	77
PACKAGE BODY	66
TYPE BODY	6
TABLE	53
INDEX	68
VIEW	81
FUNCTION	2
INDEXTYPE	4
TYPE	36

14 rows selected.

```
select object_name, object_type, status from dba_objects
where owner='CTXSYS' and status != 'VALID' order by object_name;
```

no rows selected

## 2. Context Full Text 검색

### 2.1 USER 생성

이 과정에서 사용하는 DB 계정은 'myuser' 를 사용합니다. 해당 계정을 생성하고 grant를 부여합니다. Grant를 줄때 'CTXAPP'를 확인하십시오.

```
conn / as sysdba
CREATE USER myuser IDENTIFIED BY welcome1 default tablespace users quota unlimited on users;
GRANT RESOURCE, CONNECT, CTXAPP TO myuser;
```

### 2.2 CTX PL/SQL package들을 사용할 수 있는 권한 부여

Oracle Text를 사용하기 위해서는 CTX PL/SQL을 실행 할 수 있는 권한이 필요합니다. Oracle Text를 사용하는 계정에 다음과 같이 "execute" 권한을 부여합니다.

```
GRANT EXECUTE ON CTXSYS.CTX_CLS TO myuser;
GRANT EXECUTE ON CTXSYS.CTX_DDL TO myuser;
GRANT EXECUTE ON CTXSYS.CTX_DOC TO myuser;
GRANT EXECUTE ON CTXSYS.CTX_OUTPUT TO myuser;
GRANT EXECUTE ON CTXSYS.CTX_QUERY TO myuser;
GRANT EXECUTE ON CTXSYS.CTX_REPORT TO myuser;
GRANT EXECUTE ON CTXSYS.CTX_THES TO myuser;
GRANT EXECUTE ON CTXSYS.CTX_ULEXER TO myuser;
```

### 2.3 실습용 Table 생성

아래와 같이 실습에 사용할 table을 생성합니다.

```
conn myuser/welcome1
CREATE TABLE docs (id NUMBER PRIMARY KEY, text VARCHAR2(200));
```

## 2.4 데이터 삽입

docs table에 데이터를 삽입합니다.

```
INSERT INTO docs VALUES(1, 'California is a state in the US.');
```

```
INSERT INTO docs VALUES(2, 'Paris is a city in France.');
```

```
INSERT INTO docs VALUES(3, 'France is in Europe.');
```

```
INSERT INTO docs VALUES(4, '서울은 한국의 수도입니다.');
```

```
INSERT INTO docs VALUES(5, '오라클은 삼성동에 위치하고 있습니다.');
```

지금까지 Oracle Text를 사용하기 위한 package 설치 및 색인 대상 Table을 생성했습니다. 다음 단계는 Index(색인)DB를 만들기 위한 실습을 진행합니다.

## 2.5 Context 색인 DB 생성

이번 세션에서는 Full-text 검색을 위한 Context Index DB를 생성하는 절차를 진행합니다. 색인 대상은 table의 'Text' column을 대상으로 합니다.

따라서 Datastore는 "Default\_datastore"로 지정합니다. 'text' column에 들어 있는 문자들은 plain text이기 때문에 filter를 사용할 필요가 없어 "NULL\_FILTER"를 사용하며, 한글을 word(token) 단위로 분리하기 위해서 "KOREAN\_MORPH\_LEXER" lexer를 사용합니다.

- context 인덱스를 생성하기 위해 다음과 같이 실행합니다.

```
exec ctx_ddl.drop_preference('mylexer');
```

```
exec ctx_ddl.create_preference('mylexer','KOREAN_MORPH_LEXER');
```

```
CREATE INDEX idx_docs ON docs(text)
  INDEXTYPE IS CTXSYS.CONTEXT PARAMETERS
  ('DATASTORE ctxsys.default_datastore FILTER ctxsys.null_filter LEXER mylexer');
```

-----

Index created.

- Oracle Text의 Index는 테이블에 생성되며, 인덱스 테이블과 원 테이블 간의 reference는 중간 mapping table을 통해서 이루어집니다. CONTEXT 인덱스는 인덱스 생성시 아래와 같이 4개의 관련 Table들이 생성됩니다.

table name	내용
DR\$ index_name \$I	index data를 가진 table
DR\$ index_name \$K	keymap table
DR\$ index_name \$N	negative list table
DR\$ index_name \$R	rowid 정보를 가진 table

- 참고

Oracle Text 인덱스는 inverted Index 구조로 물리적으로는 테이블 형태로 정의됩니다. Inverted Index를 사용하는 이유는 문서의 삭제 및 변경에 따른 인덱스 테이블의 변경을 효율적으로 수행하기 위해서입니다.

만일, Table에 있는 하나의 레코드가 삭제된 경우, Text Engine이 수행하는 일은 단순히 해당 Mapping Table을

삭제하는 것입니다. 삭제된 정보는 문서 검색 시에 같이 검색이 되지만, 해당 Mapping Table이 존재하지 않기 때문에 사용자에게까지 전달되지 않게 됩니다.

Mapping Table이 없는 경우, 삭제된 문서를 인덱스 테이블에 적용하기 위해서는, 먼저 모든 인덱스 테이블을 full-scan하여, 해당 문서를 포함하는 모든 token\_info를 수정해야 하는데 이는 너무나 많은 비용을 요구하는 일입니다(token\_info는 BLOB 형식임).

Inverted Index의 단점은 삭제 및 변경이 많아지면 garbage가 증가한다는 점입니다.이 문제를 해결하기 위해서는 index rebuild 또는 optimazation을 통해 주기적으로 garbage 문제를 해결해야 합니다.



#### [Inverted Index 구조]

- 관련된 Index Table들을 확인하고, Token이 들어 있는 Index table을 조회합니다. 출력 결과의 아랫 부분에는 lexer를 통해서 분석된 한글 Key word들을 확인 할 수 있습니다.

```
select * from tab where tname like 'DR%';
```

TNAME	TABTYPE	CLUSTERID
DR\$IDX_DOCS\$R	TABLE	
DR\$IDX_DOCS\$N	TABLE	
DR\$IDX_DOCS\$K	TABLE	
DR\$IDX_DOCS\$I	TABLE	

```
select token_text, token_count from DR$IDX_DOCS$I;
```

TOKEN_TEXT	TOKEN_COUNT
CALIFORNIA	1
CITY	1
EUROPE	1
FRANCE	2
IN	3
IS	3
PARIS	1
STATE	1
THE	1
US	1
삼성동	1
서울	1
수도	1
오라클	1
위치	1
한국	1

16 rows selected.

## 2.6 Full Text 검색 실행

다음은 색인 결과를 토대로 실제 Full text 검색을 실행합니다. 검색은 where 조건절에 "CONTAINS"를 사용한 select 쿼리문으로 검색을 합니다.

```
column text format a40;
SELECT SCORE(1), id, text FROM docs WHERE CONTAINS(text, 'France or 서울', 1) > 0;
```

SCORE(1)	ID	TEXT
4	2	Paris is a city in France.
4	3	France is in Europe.
5	4	서울은 한국의 수도입니다.

## 2.7 색인DB의 동기화

검색 대상인 docs table에 Insert, Update, Delete 등의 DML에 의해 데이터가 변경이 된 것을 색인 DB에 반영시키기 위해서는 "CTX\_DDL.SYNC\_INDEX" procedure를 사용하여 색인 DB를 갱신해야 합니다.

실습에서는 색인 DB인 "idx\_docs"를 2MB 메모리를 사용하여 동기화를 시킵니다. "SYNC\_INDEX"에 대한 syntax 및 세부 내용은 [Oracle Text Reference 12c Release 1 \(E41399-5\)](#)를 참조하십시오.

```
INSERT INTO docs VALUES(6, 'Los Angeles is a city in California.');
```

```
INSERT INTO docs VALUES(7, 'Mexico City is big.');
```

```
commit;
```

```
COLUMN text FORMAT a50;
```

```
SELECT SCORE(1), id, text FROM docs WHERE CONTAINS(text, 'city', 1) > 0;
```

SCORE(1)	ID	TEXT
5	2	Paris is a city in France.

```
EXEC CTX_DDL.SYNC_INDEX('idx_docs', '2M');
```

```
SELECT SCORE(1), id, text FROM docs WHERE CONTAINS(text, 'city', 1) > 0;
```

SCORE(1)	ID	TEXT
4	2	Paris is a city in France.
4	6	Los Angeles is a city in California.
4	7	Mexico City is big.

## 3. Catalog Application 검색

다음은 catalog application 검색을 위한 Catalog Index를 생성하는 예제입니다. 예제에서 다루는 Auction Table과 Catalog index의 구조는 아래 그림을 참조하십시오.

 Auction table schema and CTXCAT index

[Auction table schema and CTXCAT index 구조]

### 3.1 Table 생성 및 Data loading

"myuser"에서 다음 스크립트를 실행하여 Auction table과 실습에 활용할 데이터를 loading합니다.

```
conn myuser/welcome1;
```

```
CREATE TABLE auction(
```

```

item_id NUMBER,
title VARCHAR2(100),
category_id NUMBER,
price NUMBER,
bid_close DATE);

```

```

# Data loading

```

```

INSERT INTO AUCTION VALUES(1, 'NIKON CAMERA', 1, 400, '24-OCT-2002');
INSERT INTO AUCTION VALUES(2, 'OLYMPUS CAMERA', 1, 300, '25-OCT-2002');
INSERT INTO AUCTION VALUES(3, 'PENTAX CAMERA', 1, 200, '26-OCT-2002');
INSERT INTO AUCTION VALUES(4, 'CANON CAMERA', 1, 250, '27-OCT-2002');
commit;

```

## 3.2 Sub-index 생성

Oracle Text의 효율적인 쿼리를 위해 price column에 ordering에 사용할 sub-index를 생성합니다.

```

EXEC CTX_DDL.CREATE_INDEX_SET('auction_iset');
EXEC CTX_DDL.ADD_INDEX('auction_iset','price'); /* sub-index A */

```

## 3.3 CTXCAT index 생성 및 검색

Auction table에 Catalog Index(CTXCAT)를 생성하고, "CATSEARCH" 건을 사용해서 검색 쿼리를 수행합니다.

```

CREATE INDEX auction_titlex ON AUCTION(title) INDEXTYPE IS CTXSYS.CTXCAT PARAMETERS ('index set auction_iset');
COLUMN title FORMAT a40;
SELECT title, price FROM auction WHERE CATSEARCH(title, 'CAMERA', 'order by price')> 0;

```

TITLE	PRICE
PENTAX CAMERA	200
CANON CAMERA	250
OLYMPUS CAMERA	300
NIKON CAMERA	400

```

SELECT title, price FROM auction WHERE CATSEARCH(title, 'CAMERA',
'price <= 300')>0;

```

ITLE	PRICE
PENTAX CAMERA	200
CANON CAMERA	250
OLYMPUS CAMERA	300

## 3.4 Table update와 CTXCAT index 갱신


Auction table에 데이터를 삽입한 후, 쿼리를 수행하면 Update table의 내용이 CTXCAT index에 반영된 결과를 확인할 수 있습니다.

```
INSERT INTO AUCTION VALUES(5, 'FUJI CAMERA', 1, 350, '28-OCT-2002');
INSERT INTO AUCTION VALUES(6, 'SONY CAMERA', 1, 310, '28-OCT-2002');
commit;
```

```
SELECT title, price FROM auction WHERE CATSEARCH(title, 'CAMERA', 'order by price')> 0;
```

TITLE	PRICE
PENTAX CAMERA	200
CANON CAMERA	250
OLYMPUS CAMERA	300
SONY CAMERA	310
FUJI CAMERA	350
NIKON CAMERA	400

## 4. Classification Application 검색

Classification application Function은 Document content를 기준으로 작동하며, 문서의 category ID를 할당하거나 사용자에게 문서를 송신하는 단계를 포함 할 수 있습니다. 문서들은 선 정의된 rule들에 따라 분류되고, rule들에 의해 category가 선택합니다. 예를 들어, '대통령 선거'라는 query rule은 정치관련 category 문서들을 선택 할 수 있습니다. Oracle text는 Rule-Based classification, Supervised Classification, Unsupervised Classification(Clustering) 등의 분류(Classification) 유형을 제공합니다. 실습에서는 **rule-based** (또는 simple)분류를 사용하는데, 문서들의 분류를 위해 document categorie와 rule 을 만듭니다 

[Overview of a Document Classification Application ]

### 4.1 Rule table 생성 및 Data load

먼저 Rule Table "querues"를 만듭니다. 'queries' Rule table의 각 row는 category ID와 쿼리 문자열의 Rule을 의미합니다.

```
connect myuer/welcome1;

CREATE TABLE queries (
    query_id      NUMBER,
    query_string  VARCHAR2(80)
);

# Data load
INSERT INTO queries VALUES (1, 'oracle');
INSERT INTO queries VALUES (2, 'larry or ellison');
INSERT INTO queries VALUES (3, 'oracle and text');
INSERT INTO queries VALUES (4, 'market share');
commit;
```

### 4.2 CTXRULE index 생성

Rule table의 쿼리 문자열이 있는 Column에 'CTXRULE' 인덱스를 생성합니다.

```
CREATE INDEX queryx ON queries(query_string) INDEXTYPE IS CTXSYS.CTXRULE;
```



## 4.3 MATCHES 연산자를 이용한 Category ID 검색

'MATCHES' 연산자를 이용하여 검색을 합니다. 'MATCHES' 연산에 의해 Documents(문자열) 내용 중에 Rule로 정의된 값과 일치하는 Category ID 값을 확인 할 수 있습니다.

```
COLUMN query_string FORMAT a35;
SELECT query_id,query_string FROM queries
WHERE MATCHES(query_string,'Oracle announced that its market share in databases increased over the last year')

QUERY_ID QUERY_STRING
-----
1 oracle
4 market share
```

## 5. Document Service

### Building the JSP WEB Application

#### 5.1 Text table 생성

```
conn / as sysdba
create table search_table
(tk      number primary key,
 title  varchar2(2000),
 text   clob);
// tk      number primary key,      (primary key is important for document services)
```

#### 5.2 Result table 생성

```
CREATE TABLE output_table (query_id NUMBER, document CLOB);
CREATE TABLE gist_table (query_id NUMBER, pov VARCHAR2(80), gist CLOB);
CREATE TABLE theme_table (query_id NUMBER, theme VARCHAR2(2000), weight NUMBER);
```

#### 5.3 Load HTML Document

```
% sqlldr userid=scott/tiger control=loader.ctl
```

##### 5.3.1 loader.ctl sample

```
LOAD DATA
  CHARACTERSET UTF8
  INFILE 'loader.dat'
  INTO TABLE search_table
  replace
  FIELDS TERMINATED BY ';' optionally enclosed by '"'
  (tk          INTEGER,
   title       CHAR,
```

```
text_file    FILLER CHAR,  
text         LOBFIL(text_file) TERMINATED BY EOF)
```

### 5.3.2 loader.dat sample



```
1; Pizza Shredder;./data/Pizza.html  
2; Refrigerator w/ Front-Door Auto Cantaloupe Dispenser;./data/Cantaloupe.html  
3; Self-Tipping Couch;./data/Couch.html  
4; Home Air Dirtier;./data/Mess.html  
5; Set of Pet Magnets;./data/Pet.html  
6; Esteem-Building Talking Pillow;./data/Snooze.html  
7; "BI시장 동향" ;./data/BI_introduce.html  
8; "Database Corruption 발생 및 복구" ;./data/Database_Corruption_recovery.html  
9; "정보시스템 운영관리 지침" ;./data/Information_management_Guide.html  
10; "대기업의 공공소프트웨어사업 참여제한 예외사업";./data/ministry_2013-3.html  
11; "RAC 구성 프로세스" ;./data/RAC_configuration_process.html
```

- html data file들은 ~oracle/text/data 디렉토리에 있는 것을 활용하십시오.

## 5.4 Preference & Context 인덱스 생성

```
exec ctx_ddl.create_preference('mylexer','KOREAN_MORPH_LEXER');  
exec ctx_ddl.create_section_group('mysection','HTML_SECTION_GROUP');  
  
//참고 : exec ctx_doc.set_key_type('PRIMARY_KEY');  
  
create index idx_search_table on search_table(text)  
  indextype is ctxsys.context parameters  
  ('datastore ctxsys.default_datastore filter ctxsys.null_filter section group mysection lexer mylexer');
```

## 5.5 sample code 실행

- sample code 실행은 Oracle Web logic 또는 Apache tomcat 등의 JSP 서비스를 위한 Middleware 환경에서 실행이 가능합니다.
- sample code의 화면은 다음과 같이 실행됩니다  

## 5.6 WEB Application sample code

### 5.6.1 TextSearchApp.jsp

```
<%@page language="java" pageEncoding="utf-8" contentType="text/html; charset=utf-8" %>  
<%@ page import="java.sql.*, java.util.*, java.net.*,  
    oracle.jdbc.*, oracle.sql.*" %>  
  
<%  
    // Change these details to suit your database and user details  
  
String connStr = "jdbc:oracle:thin:@//servername:1521/pdb1";  
String dbUser  = "scott";  
String dbPass  = "tiger";
```

```

// The table we're running queries against is called SEARCH_TABLE.
// It must have columns:
// tk      number primary key,      (primary key is important for document services)
// title   varchar2(2000),
// text    clob
// There must be a CONTEXT index called IDX_SEARCH_TABLE on the text column

request.setCharacterEncoding("UTF-8");

java.util.Properties info=new java.util.Properties();
Connection conn  = null;
ResultSet rset   = null;
OracleCallableStatement callStmt = null;
Statement stmt   = null;
String userQuery = null;
String myQuery   = null;
String action    = null;
String theTk     = null;
URLDecoder myEncoder;
int count=0;
int loopNum=0;
int startNum=0;

userQuery      = request.getParameter("query");
action         = request.getParameter("action");
theTk          = request.getParameter("tk");

if (action == null) action = "";

// Connect to database

try {
    DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver() );
    info.put ("user",      dbUser);
    info.put ("password", dbPass);
    conn      = DriverManager.getConnection(connStr,info);
}
catch (SQLException e) {
%>    <b>Error: </b> <%= e %><p> <%
}

if ( action.equals("doHTML") ) {
    // Directly display the text of the document
    try {

        // not attempting to share the output table for this example, we'll truncate it each time
        conn.createStatement().execute("truncate table OUTPUT_TABLE");

        String sql = "{ call ctx_doc.filter( index_name=>'IDX_SEARCH_TABLE', textkey=>'" + theTk + "', restab=>";
        PreparedStatement s = conn.prepareStatement( sql );
        s.execute();

        sql = "select document from output_table where rownum = 1";
        stmt = conn.createStatement();
        rset = stmt.executeQuery(sql);
    }
}

```

```

    rset.next();
    oracle.sql.CLOB res = (oracle.sql.CLOB) rset.getClob(1);
    // should fetch from clob piecewise, but to keep it simple we'll just fetch 32K to a string
    String txt = res.getSubString(1, 32767);
    out.println(txt);
}
catch (SQLException e) {
%>    <b>Error: </b> <%= e %><p> <%
}
}
else if ( action.equals("doHighlight") ) {
    // Display the text of the document with highlighting from the "markup" function
    try {

        // not attempting to share the output table for this example, we'll truncate it each time
        conn.createStatement().execute("truncate table OUTPUT_TABLE");

        String sql = "{ call ctx_doc.markup( index_name=>'IDX_SEARCH_TABLE', textkey=> '"' + theTk + "', text_que
        PreparedStatement s = conn.prepareStatement( sql );
        s.execute();

        sql = "select document from output_table where rownum = 1";
        stmt = conn.createStatement();
        rset = stmt.executeQuery(sql);

        rset.next();
        oracle.sql.CLOB res = (oracle.sql.CLOB) rset.getClob(1);
        // should fetch from clob piecewise, but to keep it simple we'll just fetch 32K to a string
        String txt = res.getSubString(1, 32767);
        out.println(txt);
    }
    catch (SQLException e) {
%>    <b>Error: </b> <%= e %><p> <%
    }
}

else if ( action.equals("doThemes") ) {
    // Display the text of the document with highlighting from the "markup" function
    try {

        // not attempting to share the output table for this example, we'll truncate it each time
        conn.createStatement().execute("truncate table THEME_TABLE");

        String sql = "{ call ctx_doc.themes( index_name=>'IDX_SEARCH_TABLE', textkey=> '"' + theTk + "', restab=>
        PreparedStatement s = conn.prepareStatement( sql );
        s.execute();

        sql = "select * from ( select theme, weight from theme_table order by weight desc ) where rownum <= 20";
        stmt = conn.createStatement();
        rset = stmt.executeQuery(sql);
        int    weight = 0;
        String theme = "";
%>
<h2>The top 20 themes of the document</h2>

```

```

<table BORDER=1 CELLSPACING=0 CELLPADDING=0"
  <tr bgcolor="#CCCC99">
    <th><font face="arial" color="#336699">Theme</th>
    <th><font face="arial" color="#336699">Weight</th>
  </tr>
<%
  while ( rset.next() ) {

    theme = rset.getString(1);
    weight = (int)rset.getInt(2);
%>

    <tr bgcolor="ffffe0">
      <td align="center"><font face="arial"><b> <%= theme %> </b></font></td>
      <td align="center"><font face="arial"> <%= weight %></font></td>
    </tr>
<%
  }

%>
</table>
<%
  }
  catch (SQLException e) {
%>    <b>Error: </b> <%= e %><p> <%
  }
}
else if ( action.equals("doGists") ) {
  // Display the text of the document with highlighting from the "markup" function
  try {

    // not attempting to share the output table for this example, we'll truncate it each time
    conn.createStatement().execute("truncate table GIST_TABLE");

    String sql = "{ call ctx_doc.gist( index_name=>'IDX_SEARCH_TABLE', textkey=> '" + theTk + "', restab=>'GIST_TABLE', restabkey=>'GIST_KEY', restabval=>'GIST_VAL' ) }";
    PreparedStatement s = conn.prepareStatement( sql );
    s.execute();

    sql = "select pov, gist from gist_table where pov = 'GENERIC' and query_id = 1";
    stmt = conn.createStatement();
    rset = stmt.executeQuery(sql);
    String pov = "";
    String gist = "";

    while ( rset.next() ) {

      pov = rset.getString(1);
      oracle.sql.CLOB gistClob = (oracle.sql.CLOB) rset.getClob(2);

      out.println("<h3>Document Gist for Point of View: " + pov + "</h3>");
      gist = gistClob.getSubString(1, 32767);
      out.println(gist);

    }

%>

```

```

</table>
<%
    }
    catch (SQLException e) {
%>    <b>Error: </b> <%= e %><p> <%
    }
}

if ( (action.equals("")) && ( userQuery == null ) || (userQuery.length() == 0) ) ) {
%>
    <html>
        <title>Text Search</title>
        <body>
            <table width="100%">
                <tr bgcolor="#336699">
                    <td><font face="arial" align="left"
                        color="#CCCC99" size="+2">Text Search</td>
                </tr>
            </table>
            <center>
                <form method = post>
                    Search for :
                    <input type="text" name="query" size = "30">
                    <input type="submit" value="Search">
                </form>
            </center>
        </body>
    </html>
<%
}
else if (action.equals("") ) {
%>
    <html>
        <title>Text Search Result Page</title>
        <body text="#000000" bgcolor="#FFFFFF" link="#663300"
            vlink="#996633" alink="#ff6600">
            <table width="100%">
                <tr bgcolor="#336699">
                    <td><font face="arial" align="left"
                        color="#CCCC99" size="+2">Text Search</td>
                </tr>
            </table>
            <center>
                <form method = post action="TextSearchApp.jsp">
                    Search for :
                    <input type=text name="query" value="<%= userQuery %>" size = 30>
                    <input type=submit value="Search">
                </form>
            </center>
<%
myQuery    =    URLEncoder.encode(userQuery);
try {

    stmt      = conn.createStatement();

```

```

String numStr = request.getParameter("sn");
if(numStr!=null)
    startNum=Integer.parseInt(numStr);
String theQuery = translate(userQuery);

callStmt =(OracleCallableStatement)conn.prepareCall("begin "+
    "?:=ctx_query.count_hits(index_name=>'IDX_SEARCH_TABLE', "+
    "text_query=>?" +
    "); " +
    "end: ");
callStmt.setString(2,theQuery);
callStmt.registerOutParameter(1, OracleTypes.NUMBER);
callStmt.execute();
count=((OracleCallableStatement)callStmt).getNUMBER(1).intValue();
if(count>=(startNum+20)){
%>
<font color="#336699" FACE="Arial" SIZE=+1>Results
    <%=startNum+1%> - <%=startNum+20%> of <%=count%> matches
<%
}
else if(count>0){
%>
<font color="#336699" FACE="Arial" SIZE=+1>Results
    <%=startNum+1%> - <%=count%> of <%=count%> matches
<%
}
else {
%>
<font color="#336699" FACE="Arial" SIZE=+1>No match found
<%
}
%>
<table width="100%">
<TR ALIGN="RIGHT">
<%
if((startNum>0)&(count<=startNum+20))
{
%>
<TD ALIGN="RIGHT">
<a href="TextSearchApp.jsp?sn=<%=startNum-20 %>&query=
    <%=myQuery %>">previous20</a>
</TD>
<%
}
else if((count>startNum+20)&(startNum==0))
{
%>
<TD ALIGN="RIGHT">
<a href="TextSearchApp.jsp?sn=<%=startNum+20
    %>&query=<%=myQuery %>">next20</a>
</TD>
<%
}
else if((count>startNum+20)&(startNum>0))
{

```

```

%>
<TD ALIGN="RIGHT">
<a href="TextSearchApp.jsp?sn=<%=startNum-20 %>&query=
    <%=myQuery %>">previous20</a>
<a href="TextSearchApp.jsp?sn=<%=startNum+20 %>&query=
    <%=myQuery %>">next20</a>

</TD>

<%
}
%>
</TR>
</table>

<%
String ctxQuery =
    " select /*+ FIRST_ROWS */ " +
    " tk, TITLE, score(1) scr, " +
    " ctx_doc.snippet ('IDX_SEARCH_TABLE', tk, '" + theQuery + "') " +
    " from search_table " +
    " where contains(TEXT, '"+theQuery+"',1 ) > 0 " +
    " order by score(1) desc";
rset = stmt.executeQuery(ctxQuery);
String tk = null;
String[] colToDisplay = new String[1];
int myScore = 0;
String snippet = "";
int items = 0;
while (rset.next())&&items< 20) {
    if(loopNum>=startNum)
    {
        tk = rset.getString(1);
        colToDisplay[0] = rset.getString(2);
        myScore = (int)rset.getInt(3);
        snippet = rset.getString(4);
        items++;
        if (items == 1) {
%>

<center>
<table BORDER=1 CELLSPACING=0 CELLPADDING=0 width="100%"
    <tr bgcolor="#CCCC99">
        <th><font face="arial" color="#336699">Score</th>
        <th><font face="arial" color="#336699">TITLE</th>
        <th><font face="arial" color="#336699">Snippet</th>
        <th><font face="arial"
            color="#336699">Document Services</th>
    </tr>

<%
} %>
<tr bgcolor="#FFFFE0">
    <td ALIGN="CENTER"> <%= myScore %></td>
    <td> <%= colToDisplay[0] %> </td>
    <td> <%= snippet %> </td>
    <td>
        <a href="TextSearchApp.jsp?action=doHTML&tk=<%= tk %>">HTML</a> &nbsp;
        <a href="TextSearchApp.jsp?action=doHighlight&tk=<%= tk %>&query=<%= theQuery %>">Highlight</a> &
        <a href="TextSearchApp.jsp?action=doThemes&tk=<%= tk %>&query=<%= theQuery %>">Themes</a> &nbsp;
        <a href="TextSearchApp.jsp?action=doGists&tk=<%= tk %>">Gist</a> &nbsp;

```



```

        </td>
    </tr>
<%
    }
    loopNum++;
}
} catch (SQLException e) {
%>
    <b>Error: </b> <%= e %><p>
<%
} finally {
    if (conn != null) conn.close();
    if (stmt != null) stmt.close();
    if (rset != null) rset.close();
}
%>
</table>
</center>
<table width="100%">
<TR ALIGN="RIGHT">
<%
    if((startNum>0)&(count<=startNum+20))
    {
%>
        <TD ALIGN="RIGHT">
            <a href="TextSearchApp.jsp?sn=<%=startNum-20 %>&query=
                <%=myQuery %>">previous20</a>
        </TD>
<%
    }
    else if((count>startNum+20)&(startNum==0))
    {
%>
        <TD ALIGN="RIGHT">
            <a href="TextSearchApp.jsp?sn=<%=startNum+20 %>&query=
                <%=myQuery %>">next20</a>
        </TD>
<%
    }
    else if((count>startNum+20)&(startNum>0))
    {
%>
        <TD ALIGN="RIGHT">
            <a href="TextSearchApp.jsp?sn=<%=startNum-20 %>&query=
                <%=myQuery %>">previous20</a>
            <a href="TextSearchApp.jsp?sn=<%=startNum+20 %>&query=
                <%=myQuery %>">next20</a>
        </TD>
<%
    }
%>
</TR>
</table>
</body></html>
<%}

```

```

%>
<%!
public String translate (String input)
{
    Vector reqWords = new Vector();
    StringTokenizer st = new StringTokenizer(input, " ", true);
    while (st.hasMoreTokens())
    {
        String token = st.nextToken();
        if (token.equals(" "))
        {
            String phrase = getQuotedPhrase(st);
            if (phrase != null)
            {
                reqWords.addElement(phrase);
            }
        }
        else if (!token.equals(" "))
        {
            reqWords.addElement(token);
        }
    }
    return getQueryString(reqWords);
}

private String getQuotedPhrase(StringTokenizer st)
{
    StringBuffer phrase = new StringBuffer();
    String token = null;
    while (st.hasMoreTokens() && !(token = st.nextToken()).equals(" "))
    {
        phrase.append(token);
    }
    return phrase.toString();
}

private String getQueryString(Vector reqWords)
{
    StringBuffer query = new StringBuffer("");
    int length = (reqWords == null) ? 0 : reqWords.size();
    for (int ii=0; ii < length; ii++)
    {
        if (ii != 0)
        {
            query.append(" & ");
        }
        query.append("{");
        query.append(reqWords.elementAt(ii));
        query.append("}");
    }
    return query.toString();
}
%>

```

---

# 실습 정리

---

모든 실습 과정이 끝나셨습니다. 수고하셨습니다.

```
conn / as sysdba
drop user myuser cascade;
```

[참고자료]

1. [Oracle® Text Reference 12c Release 1 \(12.1\) E41399-05](http://docs.oracle.com/database/121/CCREF/toc.htm)

[<http://docs.oracle.com/database/121/CCREF/toc.htm>]

2. [Oracle® Text Application Developer's Guide 12c Release 1 \(12.1\) E41398-05](http://docs.oracle.com/database/121/CCAPP/toc.htm)

[<http://docs.oracle.com/database/121/CCAPP/toc.htm>]

3. [Getting Started with Oracle Text](http://docs.oracle.com/database/121/CCAPP/quicktour.htm#CCAPP0200) [<http://docs.oracle.com/database/121/CCAPP/quicktour.htm#CCAPP0200>]