WITHIN You and ABOUT You:
Getting Started with interMedia Text

DOUG S.

NYOUGG December 2000

Carol Brennan, Comedy Central
Douglas Scherer, Core Paradigm
Topics to be Discussed

- Overview of interMedia Text
- Using interMedia Text
  - Create and load database tables
  - Create interMedia Text indexes on database tables
  - Search indexed documents
  - Maintain text indexes
- Gotchas
Overview of interMedia Text

- Oracle8i’s interMedia Text provides a set of extensions to standard SQL that enable powerful text searches.
  - Searches can be performed against simple types (such as VARCHAR2s)
  - Or can be performed against extended types (such as stored Word documents)
- It extends and simplifies the functionality of Oracle ConText, an add-on product available with Oracle7.
- Determine which version of interMedia Text is installed (as CTXSYS or a DBA)

```
SELECT * FROM CTXSYS.ctx_version;
```
Example of a Text Query

- interMedia Text Query

```
SELECT *
FROM emp
WHERE CONTAINS (employee_review, 'great job') > 0;
```

- Versus

```
SELECT *
FROM emp
WHERE UPPER(employee_review) LIKE '%GREAT JOB%';
```

```
SELECT *
FROM emp
WHERE INSTR(UPPER(employee_review), 'GREAT JOB') > 0;
```
Score!

- Gerald Salton
- Score = 3f(1+log(N/n))
  - f = frequency of term in the document
  - N = the total number of rows in the table
  - n = number of rows which contain the search term
- Score is converted to integer
- Final score range: 0 – 100
Score Example

Score = 3f(1+log(N/n))

Given 32000 Rows - 3200 of which contain at least one occurrence of the desired word:

- For rows containing listing the test word once
  
  (3 * 1) + (3 * 1 * log(32000/3200))
  
  = 3 + (3 * log(10))
  
  = 3 + (3 * 1) = 3 + 3 = 6

- For rows containing the test word five times
  
  (3 * 5) + (3 * 5 * log(32000/3200))
  
  = 15 + (15 * log(10))
  
  = 15 + (15 * 1) = 15 + 15 = 30

- For rows not containing the word
  
  (3 * 0) + (3 * 0 * log(32000/3200))
  
  = (3 * 0) + (3 * 0 * log(10))
  
  = 0 + 0 = 0
Create and Load Database Tables

- Create
  - CREATE TABLE statement
  - Table must have a primary key

- Load
  - INSERT statements
  - SQL*LOADER
  - import/export
  - DBMS_LOB package
CREATE THE TABLE RECIPES

(id INTEGER PRIMARY KEY,
 name VARCHAR2(100) NOT NULL,
 prep_time_minutes NUMBER,
 servings NUMBER,
 description VARCHAR2(1000),
 html_page CLOB DEFAULT EMPTY_CLOB());
Create Text Indexes

- A text index is a special index for use by interMedia Text searches.
- A text index can only be defined on one table column.
- However... more than one text index can be defined on a table.
- Syntax

```
CREATE INDEX <index_name>
  ON <table_name> (<column_name>)
  INDEXTYPE IS CTXSYS.CONTEXT
  [PARAMETERS (<'ParameterString'>)]);
```
Sizing

- Total interMedia Text size can be from 30% - 200% of size of indexed information.

- To save space
  - Set INDEX_THEMES to NO in the BASIC_LEXER (if you’re not going to use theme searches)
  - Enhance the STOP WORD list
Sizing

To get size of existing interMedia Text objects for an existing index

- Handle LOB related segments for dr$<INDEXNAME>i.token_info and dr$<INDEXNAME>r.data

```sql
SELECT sum(bytes)
FROM user_segments
WHERE segment_name LIKE 'DR$<INDEXNAME>%'
OR segment_name IN
  (SELECT segment_name
   FROM user_lobs
   WHERE table_name LIKE 'DR$<INDEXNAME>%'
  );
```
Managing Storage

- Manage storage with `BASIC_STORAGE` object in the `CREATE INDEX` statement.
- `DR$<INDEXNAME>$I,$K,$R,$N`

```sql
CREATE INDEX products_desc
ON products (description)
INDEXTYPE IS CTXSYS.CONTEXT
PARAMETERS ('STORAGE CTXSYS.PRODUCTS_STORAGE');
```
The Four Stages of Text Indexing

1. **Table**
   - Rows of data

2. **Datastore**
   - Document Data

3. **Filter**
   - Text Representation

4. **Sectioner**
   - Plain Text

5. **Lexer**
   - “Indexed” Data

6. **Text Index**
   - Text Representation
Things that Can Be Indexed

- CHAR
- VARCHAR
- VARCHAR2
- LONG
- LONG RAW
- BLOB
- CLOB
- BFILE
- URLs
- Legacy rows of data
- Custom values leading to XML fields, synthesized documents
Perform Queries Against Indexed Text

- Returns documents that contain a match for an exact word or phrase, or for a combination of exact words or phrases

- Allows for
  - Single-word match
  - Phrase Match
  - Match containing Boolean operators
  - Scoring
  - Weighted match
  - Complex Queries
Perform Queries Against Indexed Text: Single-Word

- CONTAINS clause matches one word to the text
- Example

```sql
SELECT id
FROM recipes
WHERE CONTAINS (description, 'bean') > 0;
```
Perform Queries Against Indexed Text: Phrase

- CONTAINS clause used to search for a phrase
- Example

```
SELECT id
  FROM recipes
WHERE CONTAINS (description, 'black bean soup') > 0;
```
Perform Queries Against Indexed Text: BOOLEAN

- AND, OR, and NOT used with words and phrases
- Examples

```sql
SELECT id
FROM recipes
WHERE CONTAINS (description,
    '(bean {AND} soup) OR rice'
) > 0;

SELECT id
FROM recipes
WHERE CONTAINS (description, 'bean NOT soup') > 0;
```
Perform Queries Against Indexed Text: Scoring

- Using CONTAINS return value to be used in sorting result set
- Must add a numerical argument
  - “Contains Label”
  - Represents the SCORE in CONTAINS to the rest of the statement
- Example
  ```sql
  SELECT id, SCORE(1)
  FROM recipes
  WHERE CONTAINS(description, 'bean', 1) > 0
  ORDER BY SCORE(1) DESC;
  ```
Perform Queries Against Indexed Text: Weighted

- Search terms can be assigned different weights

Example

```sql
SELECT id
  FROM recipes
 WHERE CONTAINS(description, '(bean*2) AND rice', 1) > 0
 ORDER BY SCORE(1);
```

<table>
<thead>
<tr>
<th>Doc</th>
<th>Bean</th>
<th>Rice</th>
<th>Bean and rice</th>
<th>Bean*2 and rice</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>10</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>B</td>
<td>30</td>
<td>10</td>
<td>30</td>
<td>60</td>
</tr>
<tr>
<td>C</td>
<td>20</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
</tbody>
</table>
Perform Queries Against Indexed Text: Complex Queries

In addition to the simple queries used in the previous examples, CONTAINS can be used in:

- Complex queries
- View definitions
- PL/SQL
- DML

Example

```
SELECT id
FROM recipes
WHERE CONTAINS(name, 'vegetarian') > 0
AND id NOT IN
  (SELECT id
   FROM recipes
   WHERE CONTAINS(description, 'microwave') > 0
  );
```
Perform Queries Against Indexed Text: WITHIN Clause

- Requires that documents have defined sections
- Example
  - Consider that the recipes.html_page column contains an HTML page
  - The HTML page has a section defined as `<TITLE>...</TITLE>`
  - The following search could be performed

```sql
SELECT id
FROM recipes
WHERE CONTAINS(html_page, 'stew WITHIN title') > 0;
```
Perform Queries Against Indexed Text: ABOUT Clause

- Returns documents having a similar theme as the search term
- By default, based on the interMedia Text built-in thesaurus
- If desired, this built-in thesaurus can be expanded
Perform Queries Against Indexed Text: ABOUT Clause

- Example

```
SELECT id
  FROM recipes
WHERE CONTAINS (html_page, 'ABOUT(bean)') > 0;
```
Maintain Text Indexes

- Text indexes are not updated automatically when their underlying data changes.
- They must be synchronized periodically.
Maintain Text Indexes: Delayed vs. Immediate Effects of DML

- **INSERT**: The inserted document will not be included in text search results until an index synchronization occurs.
- **DELETE**: The document is immediately excluded from text search results.
- **UPDATE**: The old version of the document is immediately excluded from text search results. The new version will not be included until an index synchronization occurs.
Maintain Text Indexes: Manual Text Index Synchronization

- Syntax
  
  ```sql
  ALTER INDEX <index_name>
  REBUILD [PARAMETERS ('SYNC')];
  ```

  “PARAMETERS('SYNC')” indicates that only changed records should be synchronized. If this is omitted, entire index is rebuilt.
Maintain Text Indexes: Jobs

- Automatic Text Index Synchronization
  - DBMS_JOB or cron
    - Schedule automatic execution of an “ALTER INDEX REBUILD…” statement
- Oracle Enterprise Manager
  - Schedule index synchronization within the Oracle Enterprise Manager Job Queue
Maintain Text Indexes: 8.1.6/7

- CTXSRV program is depreciated
- Use the following new APIs called from a DBMS_JOB
  - Put all indexes in one job
  - Put individual index per job
  - Combination of both
- For Synchronization Use CTX_DDL.SYNC_INDEX
- For Optimization Use CTX_DDL.OPTIMIZE_INDEX
Tuning

- Set INDEX THEMES to NO in the BASIC LEXER (if you’re not going to use theme searches)
- Enhance the STOPWORD list
- See Appendix A of interMedia Text documentation
- Analyze table
Tuning (continued)

- Create synthetic document to avoid b*tree and context search
- Specify NOLOGGING in storage preference for index creation
- Analyze tables that use interMedia Text indexes
Gotchas

- Temporary files in NT do not get cleaned up
- INSO_FILTER does not work properly when in NT there are two ORACLE_HOMEs.
  - Must manually configure the Filter preference
- Bug 1249652: 8.1.6 – IMP must be run as owner of interMedia Text table
  - FROMUSER TOUSER will not work with interMedia Text
Common problems

- Can tns ping, but can’t create an interMedia Text index
- Receive “ORA-06520 PL/SQL Error loading external library” upon index creation
- Receive “DRG-50704 Net8 listener not running or cannot start external procedures” when creating interMedia Text index

Configure listener.ora and tnsnames.ora for use of PLSExtProc SID_NAME

(ENVS=LD_LIBRARY_PATH=<SameAsLD_LIBRARY_PATH>:<CTXLibraryPath>)
Good Metalink Documents

- Doc ID: Note:101493.1
  - Subject: QUICK START GUIDE: interMedia Text Installation

- Doc ID: Note:92291.1
  - Subject: CBO always used when interMedia index exists (even without statistics)

- Doc ID: Note:76523.1
  - Subject: interMedia Text FAQ
Author Contact Information

- Carol Brennan
  - cbrennan@comedycentral.com

- Douglas Scherer
  - http://www.coreparadigm.com
  - dscherer@coreparadigm.com
  - Oracle8i Tips & Techniques
    - Osborne McGraw-Hill,
  - Oracle DBA Interactive Workbook and Video Course
    - Prentice Hall: ISBN 0130157422 and 0130321230