Αδόμητες ΒΔ – Oracle Text

- Review Oracle Text basics
- Index Options
- Compare Oracle Text with interMedia and ConText
- Labs - create different kinds of indexes, see what the database looks like, and query to see the impact of each change

Βασική πηγή διαφανειών: Oracle Corporation
Εργαστήριο Πληροφοριακών Συστημάτων, Παν/μιο Πειραιώς (http://infolab.cs.unipi.gr/)
έκδοση: Ιανουάριος 2010

Overview

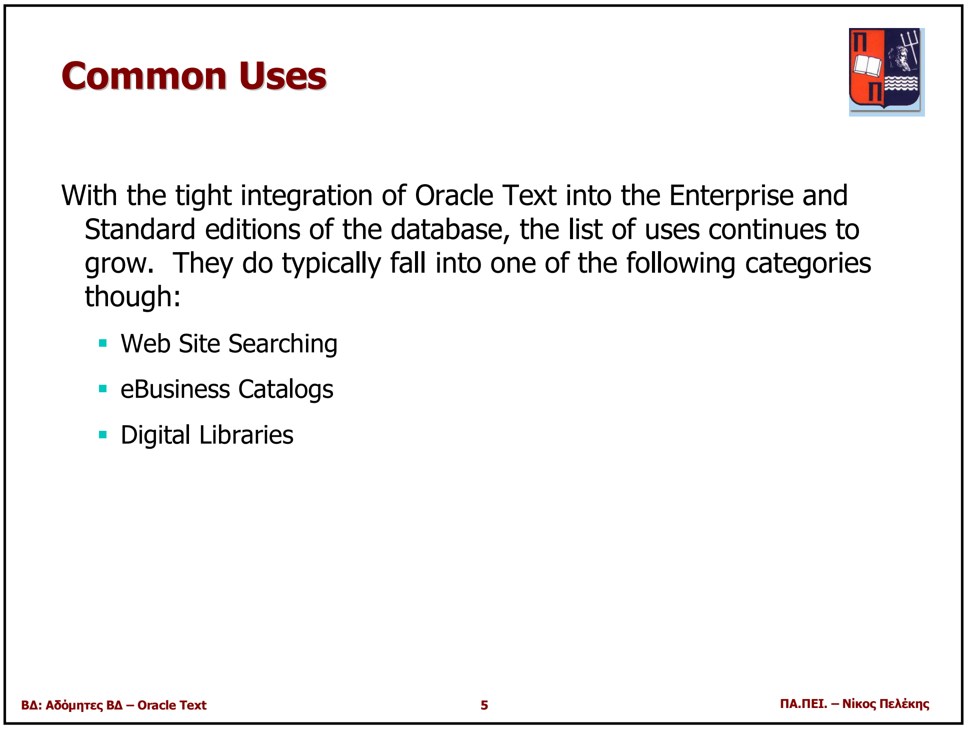
- Oracle Text adds text search and intelligent text management to 9i. It is included with both 9i Enterprise and Standard editions, and is selected as an option at db creation.
- It supports more than 150 document formats including MS Office file formats, Adobe PDF, HTML and XML.
- Searches across documents in western languages (English, French, Spanish, etc.) as well as Japanese, Korean, Traditional and Simplified Chinese.

Overview

- Oracle Text is the foundation for text processing in IFS, Oracle Ultra Search, Oracle eBusiness Suite, Oracle 9iAS Portal, and on oracle.com
- Oracle Text index management can be done using Oracle Enterprise Manager
- The SQL API allows developers and administrators to develop and maintain indexes with standard SQL syntax

Terminology

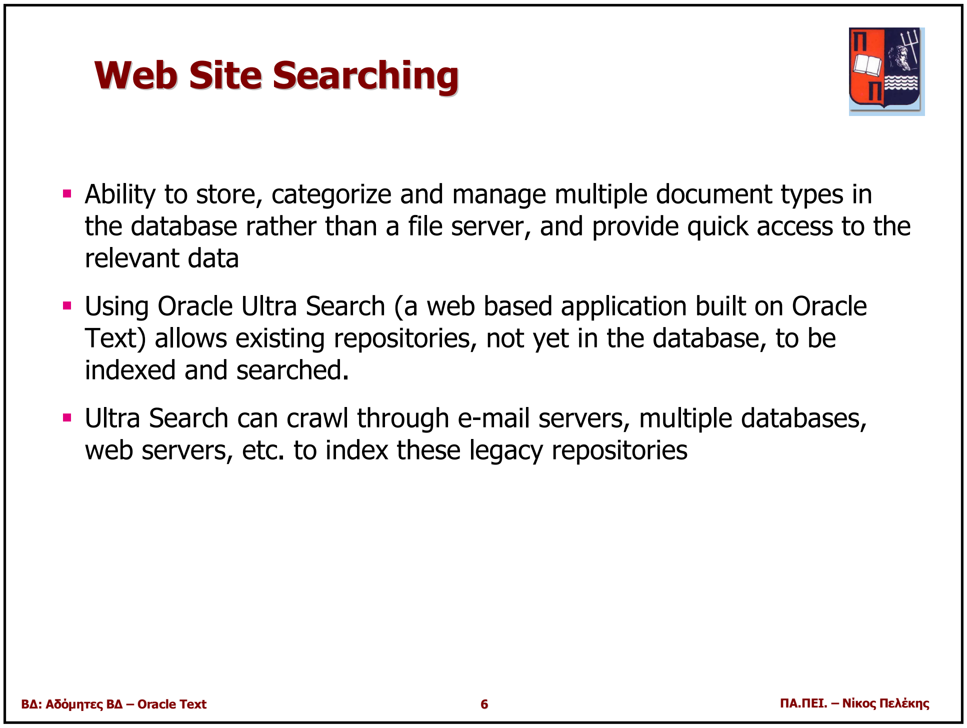
Datastore	How are your documents stored?
Filter	How can the documents be converted to plain text for indexing – INSO filter handles much of the filtering for the ~150 supported document types.
Lexer (object)	The lexer breaks the text into tokens according to your language – covered in more detail later…
Wordlist	How should stem and fuzzy queries be expanded?
Stop List	What words or themes are not to be indexed (examples could be 'of', 'the', 'if', etc)?



Common Uses

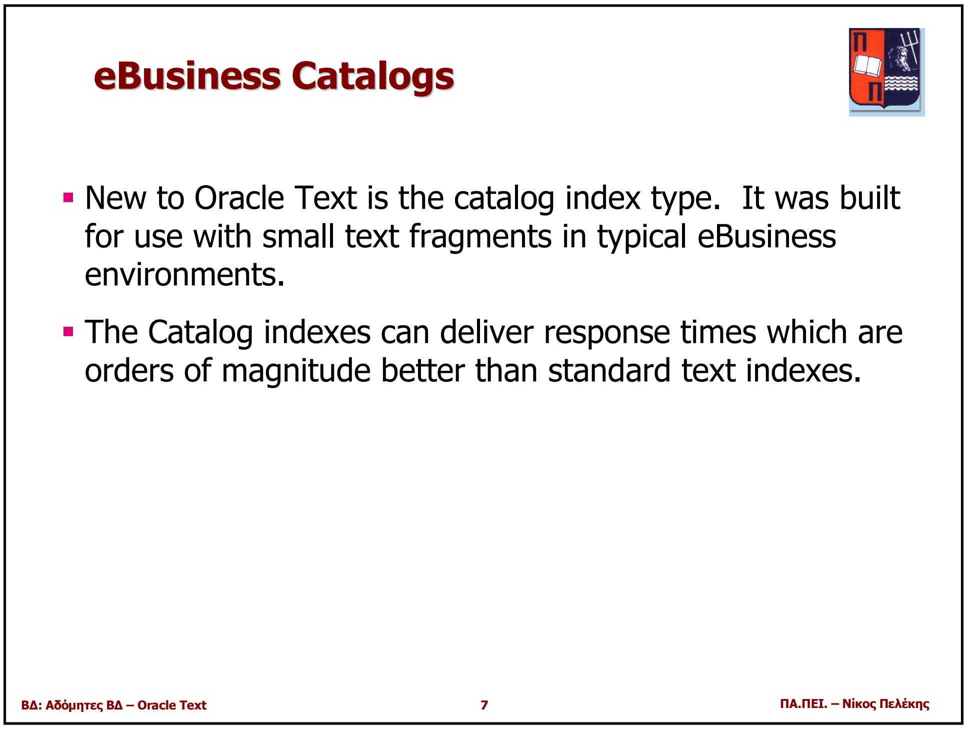
With the tight integration of Oracle Text into the Enterprise and Standard editions of the database, the list of uses continues to grow. They do typically fall into one of the following categories though:

- Web Site Searching
- eBusiness Catalogs
- Digital Libraries



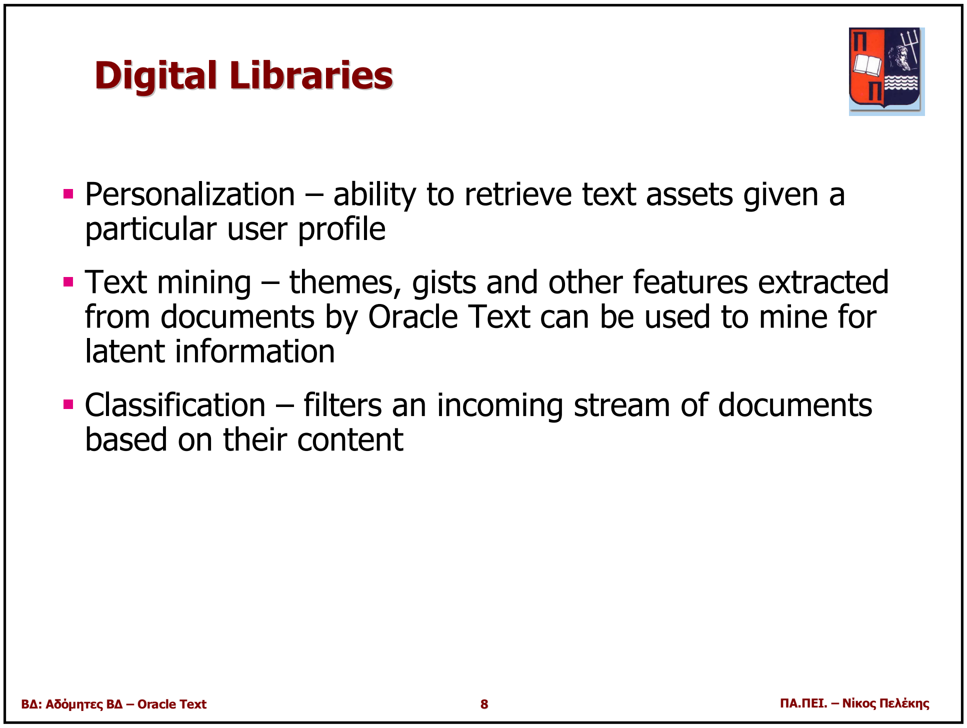
Web Site Searching

- Ability to store, categorize and manage multiple document types in the database rather than a file server, and provide quick access to the relevant data
- Using Oracle Ultra Search (a web based application built on Oracle Text) allows existing repositories, not yet in the database, to be indexed and searched.
- Ultra Search can crawl through e-mail servers, multiple databases, web servers, etc. to index these legacy repositories



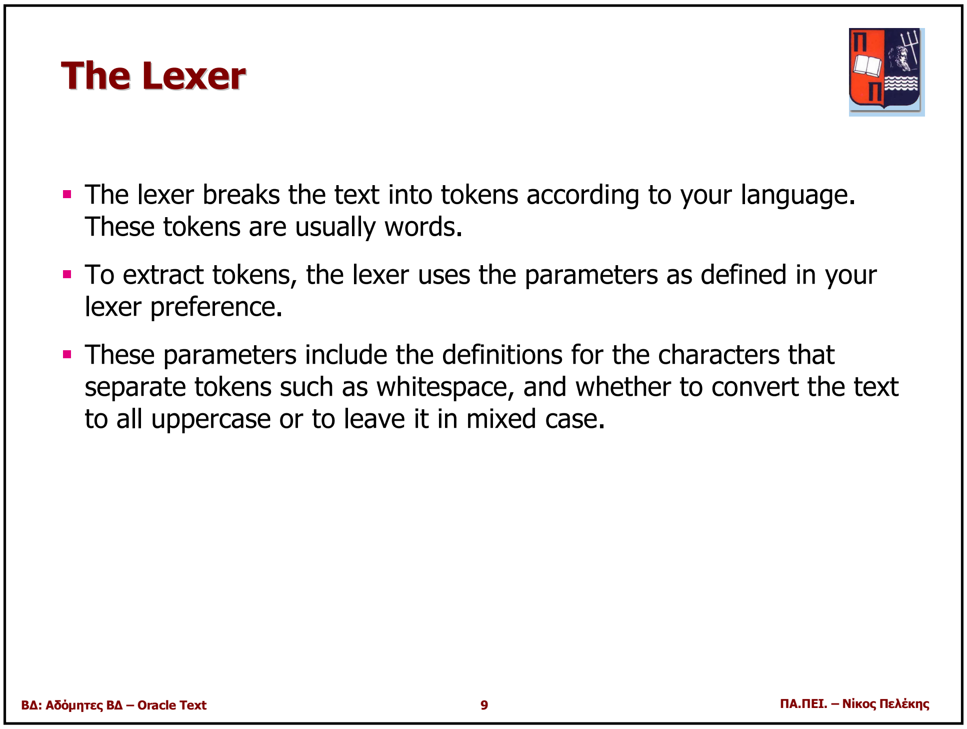
eBusiness Catalogs

- New to Oracle Text is the catalog index type. It was built for use with small text fragments in typical eBusiness environments.
- The Catalog indexes can deliver response times which are orders of magnitude better than standard text indexes.



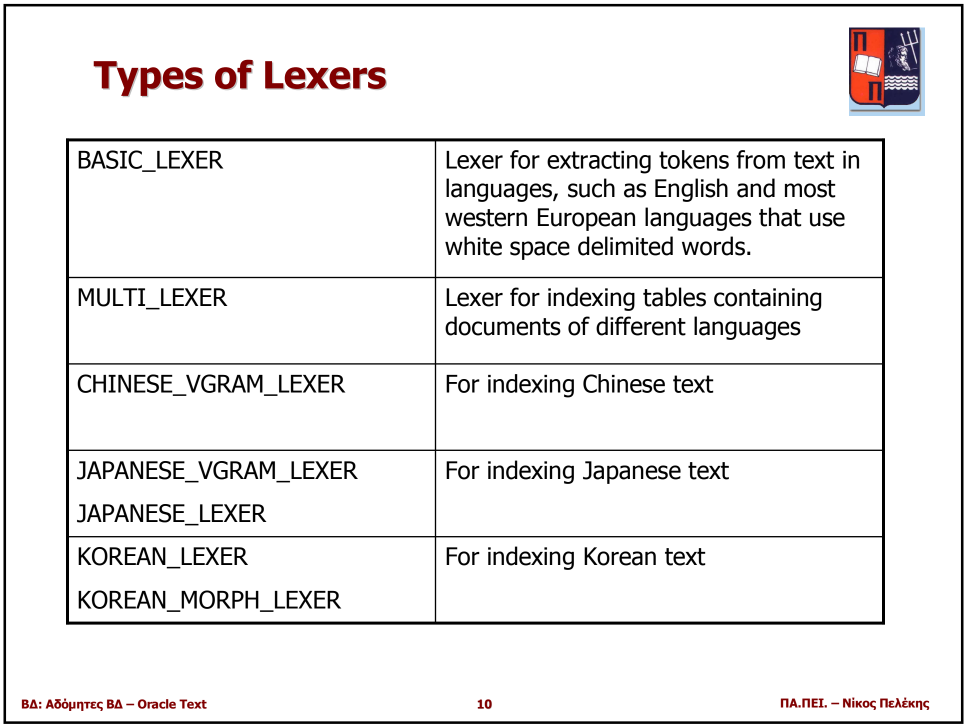
Digital Libraries

- Personalization – ability to retrieve text assets given a particular user profile
- Text mining – themes, gists and other features extracted from documents by Oracle Text can be used to mine for latent information
- Classification – filters an incoming stream of documents based on their content



The Lexer

- The lexer breaks the text into tokens according to your language. These tokens are usually words.
- To extract tokens, the lexer uses the parameters as defined in your lexer preference.
- These parameters include the definitions for the characters that separate tokens such as whitespace, and whether to convert the text to all uppercase or to leave it in mixed case.



Types of Lexers

BASIC_LEXER	Lexer for extracting tokens from text in languages, such as English and most western European languages that use white space delimited words.
MULTI_LEXER	Lexer for indexing tables containing documents of different languages
CHINESE_VGRAM_LEXER	For indexing Chinese text
JAPANESE_VGRAM_LEXER JAPANESE_LEXER	For indexing Japanese text
KOREAN_LEXER KOREAN_MORPH_LEXER	For indexing Korean text

Oracle Text Indexes

CONTEXT	Standard index	Traditional full-text retrieval over documents and web pages.
CTXCAT	Catalog index	First text index designed specifically for eBusiness catalogs.
CTXRULE	Classification index	For building classification or routing applications.
CTXXPATH	Xpath index	Improves performance on Xpath searches on XML documents.

Anatomy of the CONTEXT Index

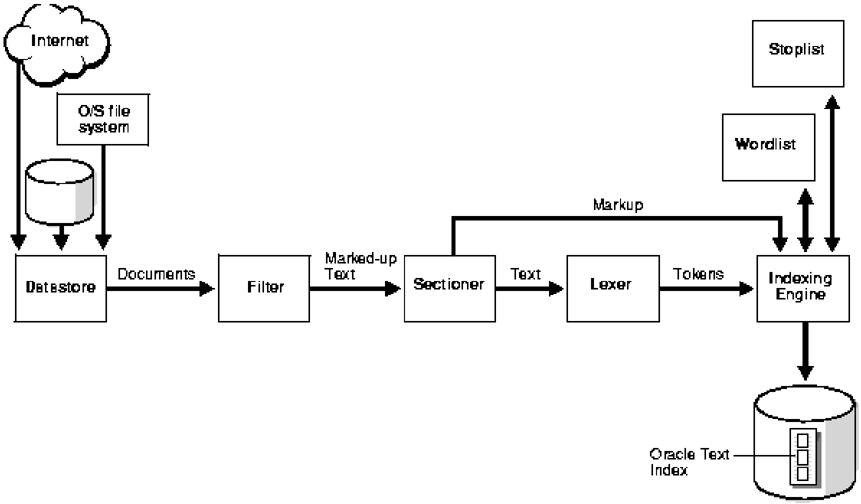
Oracle Text indexes text by converting all words into tokens. The general structure of an Oracle Text CONTEXT index is an inverted index where each token contains the list of documents (rows) that contain that token.

For example, after a single initial indexing operation, the word DOG might have an entry as follows:

```
DOG DOC1 DOC3 DOC5
```

This means that the word DOG is contained in the rows that store documents one, three and five.

Anatomy of the CONTEXT Index cont.



Index Tables

BASIC_STORAGE creates five index tables against each table indexed.

DR$...$I	Index data table. Includes all tokens that have been indexed, together with a binary representation of the documents they occur in, and their position in the documents. Each document has an internal DOCID value.
DR$...$K	Keymap table. Index-Organized Table which maps internal DOCID values to external ROWID values.
DR$...$N	Negative list table. Contains a list of deleted DOCID values – used and cleaned up by the index optimization process.
DR$...$P	Parameter clause for the substring index if you have enabled SUBSTRING_INDEX in the BASIC_WORDLIST.
DR$...$R	ROWID table. The reverse of the K table – fetching ROWID when you know the DOCID.

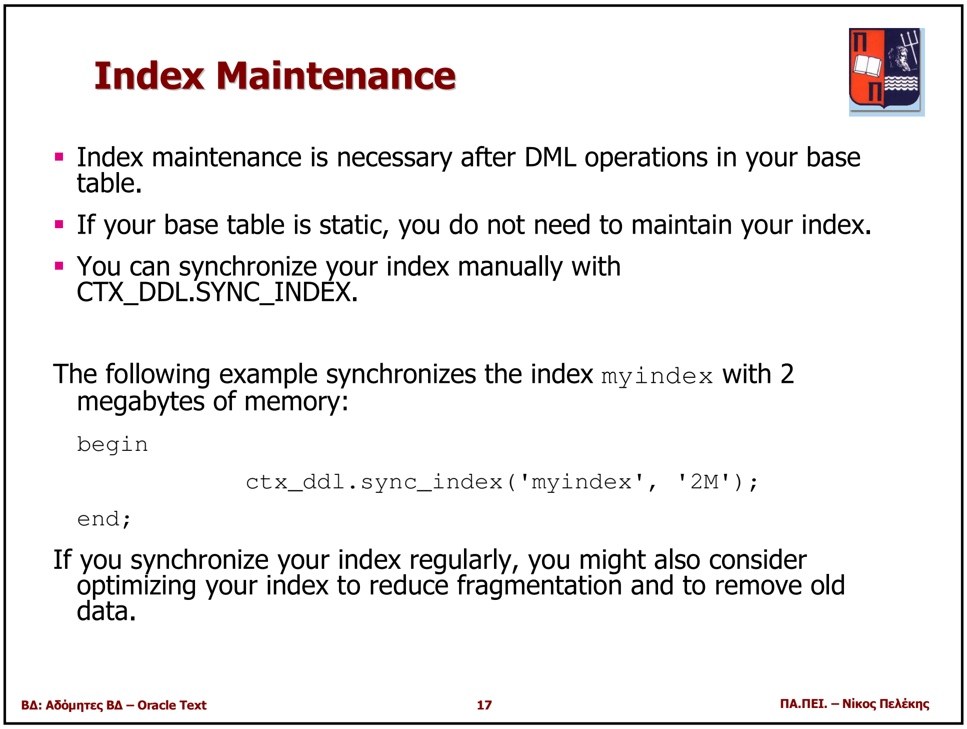
Storage Location Options

- Database – Index any text or lob column up to 4 Gig
- URL Reference – Text that can be referenced by a URL on any internet or intranet site
- File System – Any supported document type stored on a file system accessible from the db server
- User-Defined – Index the output of a PL/SQL procedure

CONTEXT Supported Datatypes

You can create a CONTEXT index with columns of the following types:

- VARCHAR2
- CLOB
- BLOB
- CHAR
- BFILE
- XMLType



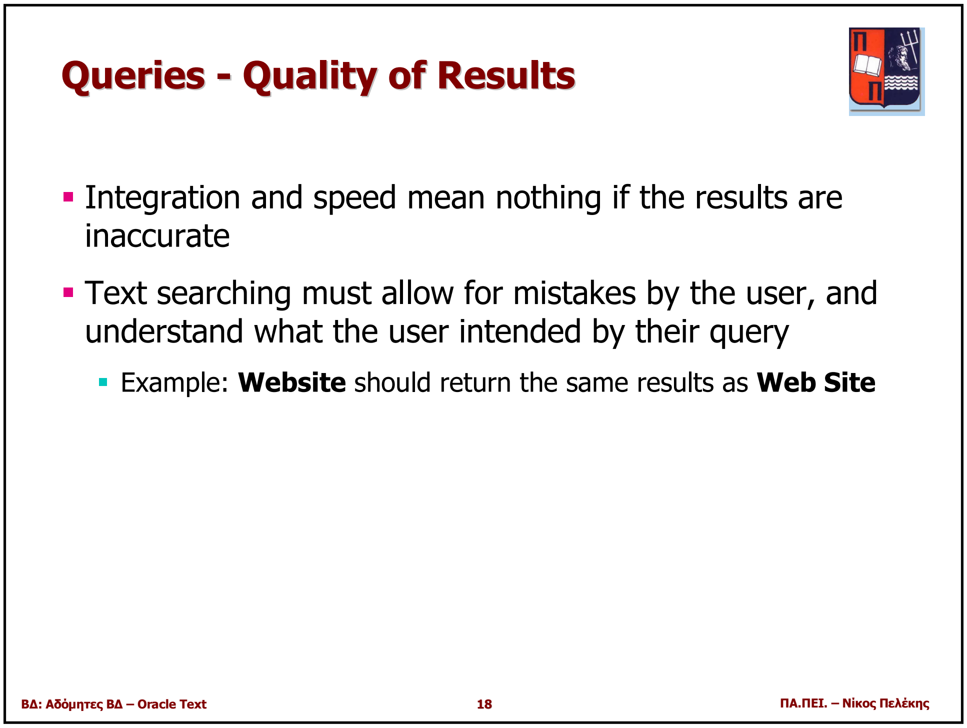
Index Maintenance

- Index maintenance is necessary after DML operations in your base table.
- If your base table is static, you do not need to maintain your index.
- You can synchronize your index manually with CTX_DDL.SYNC_INDEX.

The following example synchronizes the index `myindex` with 2 megabytes of memory:

```
begin
            ctx_ddl.sync_index('myindex', '2M');
end;
```

If you synchronize your index regularly, you might also consider optimizing your index to reduce fragmentation and to remove old data.



Queries - Quality of Results

- Integration and speed mean nothing if the results are inaccurate
- Text searching must allow for mistakes by the user, and understand what the user intended by their query
  - Example: **Website** should return the same results as **Web Site**

Supported Types of Searches

Exact Match	Enter one or more keywords that are contained in the document
Word Positioning	Search for a phrase, words near each other, or words in the same sentence
Inexact Match	Fuzzy, soundex, auto-stem, auto-wild, and thesaurus searches
Intelligent Match	Search on themes of documents while ignoring noise words
Boolean Combinations	Use of AND, OR and NOT
Relevance Ranking	Sort results according to how well the text matches the search criteria

Text Management

- Search by theme – find documents ‘about’ something
- Get the gist of a document – what sentences or paragraphs contribute most to the theme
- Load a custom thesaurus to improve search and theme extraction (example: auto manufacturing company loading a thesaurus containing standard terms)
- Associate terms to catalog items to have those items return, even when the associated terms do not exist in the indexed record itself

Query Overview

- Oracle returns all documents (previously indexed) that satisfy the expression along with a relevance score for each document. Scores can be used to order the documents in the result set.
- To issue an Oracle Text query, use the SQL SELECT statement with either the CONTAINS or CATSEARCH operator. You can use these operators programatically wherever you can use the SELECT statement, such as in PL/SQL cursors.
- Use the MATCHES operator to classify documents with a CTXRULE index.

Querying with CONTAINS

- When you create an index of type `context`, you must use the CONTAINS operator to issue your query.
- With the CONTAINS operator, you can use a number of operators to define your search criteria. These operators enable you to issue logical, proximity, fuzzy, stemming, thesaurus and wildcard searches.
- With CONTAINS, you can also use the ABOUT operator to search on document themes.

CONTAINS Example

In the SELECT statement, specify the query in the WHERE clause with the CONTAINS operator. Also specify the SCORE operator to return the score of each hit in the hitlist.

The following example shows how to issue a query:

```
SELECT SCORE(1), title
     FROM news
     WHERE CONTAINS(text, 'oracle', 1) > 0;
```

You can order the results from the highest scoring documents to the lowest scoring documents using the ORDER BY clause as follows:

```
SELECT SCORE(1), title
     FROM news
     WHERE CONTAINS(text, 'oracle', 1) > 0
     ORDER BY SCORE(1) DESC;
```

Querying with CATSEARCH

- When you create an index of type `ctxcat`, you must use the CATSEARCH operator to issue your query.
- The operators available for CATSEARCH queries are limited to logical operations such as AND or OR. The operators you can use to define your structured criteria are greater than, less than, equality, BETWEEN, and IN.
- A typical query with CATSEARCH :

```
SELECT FROM auction
       WHERE CATSEARCH(title, 'camera',
       'order by bid_close desc')> 0;
```

Oracle Ultra Search

Ultra Search is a new search application built on top of Oracle Text. It offers:

- Uniform search in a database, on the internet or in an application
- Crawling, indexing and making searchable and entire corporate intranet
- Browser-based administration interface

TEXT GROUP - Japanese Lexer Exercise

When you specify JAPANESE_LEXER for creating text index, the JAPANESE_LEXER resolves a sentence into words.
For example, the following compound word (natural language institute):

'自然言語処理'

is indexed as three tokens:

'自然', '言語', '処理'

HOW does Oracle resolve a sentence into words?

Dev Tools For Oracle Text

- Enterprise Manager – Adminstrative tools
- JDeveloper 9i – Some functionality built-in to Jdeveloper…additional add-ins scheduled for release on OTN including Catalog and Oracle Text wizards
- Add-in for Dreamweaver UltraDev/MX

Summary

- Very flexible
- Oracle’s investment in this technology clearly points toward even greater integration in applications and tools
- NLS support among the best in the industry

Sample Code

```
-- test table

CREATE TABLE media_information (
media_information_id NUMBER(10) PRIMARY KEY,
long_description VARCHAR2(4000));

INSERT INTO media_information
    VALUES (30, 'Ink leaked out of my blue pen and stained my shirt');
INSERT INTO media_information
    VALUES (31, 'A Pencil is much easier to correct than a pen');
INSERT INTO media_information
    VALUES (32, 'The preferred pencil is a NO 2');
INSERT INTO media_information
    VALUES (33, 'I need a Mouse for my computer');
INSERT INTO media_information
    VALUES (34, 'Our company uses many Mice');

BEGIN
  ctx_ddl.create_preference ('my_lexer', 'basic_lexer');
  ctx_ddl.set_attribute ('my_lexer', 'index_text', 'true');
  ctx_ddl.set_attribute ('my_lexer', 'index_themes', 'false');
END;
/

BEGIN
  ctx_ddl.create_preference ('my_wordlist', 'basic_wordlist');
  ctx_ddl.set_attribute ('my_wordlist', 'substring_index', 'true');
END;
/
```

Sample Code cont.

```
-- Create an index with an empty stoplist
CREATE INDEX media_information_indx ON media_information(long_description)
INDEXTYPE IS ctxsys.CONTEXT
PARAMETERS ( 'lexer my_lexer wordlist my_wordlist stoplist ctxsys.empty_stoplist' );

SELECT token_text
FROM DR$MEDIA_INFORMATION_INDX$I;

-- There are 32 records returned
-- Note that words such as 'A' and 'OF' are present

DROP INDEX MEDIA_INFORMATION_INDX;

-- Create an index with a stoplist

CREATE INDEX media_information_indx ON media_information(long_description)
INDEXTYPE IS ctxsys.CONTEXT
PARAMETERS ( 'lexer my_lexer wordlist my_wordlist stoplist ctxsys.default_stoplist' );

SELECT token_text
FROM DR$MEDIA_INFORMATION_INDX$I;

-- There are only 22 records returned
-- Note that words such as 'A' and 'OF' are not present
```

Sample Code cont.

```
SELECT media_information_id, long_description, score(1)
FROM media_information
WHERE CONTAINS(long_description, 'mice', 1) > 0
ORDER BY score(1);

SELECT media_information_id, long_description, score(1)
FROM media_information
WHERE CONTAINS(long_description, '$mice', 1) > 0
ORDER BY score(1);

SELECT media_information_id, long_description, score(1)
FROM media_information
WHERE CONTAINS(long_description, 'pen', 1) > 0
ORDER BY score(1);

SELECT media_information_id, long_description, score(1)
FROM media_information
WHERE CONTAINS(long_description, 'pen%', 1) > 0
ORDER BY score(1);
```

Sample Code cont.

```
-- Verify that ctx_user_pending is empty
-- Insert a new record and verify that it is NOT indexed
-- Verify that ctx_user_pending has records

select pnd_rowid,
       to_char ( pnd_timestamp, 'dd-mon-yyyy hh24:mi:ss' ) t
  from ctx_user_pending
  where pnd_index_name = 'MEDIA_INFORMATION_INDX';

INSERT INTO media_information
     VALUES (35, 'ctx pending', NULL);
COMMIT;

select pnd_rowid,
       to_char ( pnd_timestamp, 'dd-mon-yyyy hh24:mi:ss' ) t
  from ctx_user_pending
  where pnd_index_name = 'MEDIA_INFORMATION_INDX';

SELECT media_information_id, long_description, score(1)
FROM media_information
WHERE CONTAINS(long_description, 'PENDING', 1) > 0
ORDER BY score(1);
```

Sample Code cont.

```
-- Rebuild the index and verify that the value can be selected

alter index media_information_indx rebuild online parameters ( 'sync' );

select pnd_rowid,
       to_char ( pnd_timestamp, 'dd-mon-yyyy hh24:mi:ss' ) t
  from ctx_user_pending
  where pnd_index_name = 'MEDIA_INFORMATION_INDX';


SELECT media_information_id, long_description, score(1)
FROM media_information
WHERE CONTAINS(long_description, 'PENDING', 1) > 0
ORDER BY score(1);
```