„Googeln“ mit der Datenbank – Oracle Text

Dr. Frank Haney

About myself

Freelance Oracle Consultant since 2002

- OCP DBA
- Oracle University Certified Trainer
- TÜV Auditor for IT Security

Administration

- Implementation and Test
- High Availability
- Backup and Recovery
- Migration and Upgrade
- Performance Diagnostics and Tuning
- Data Security

Development

- Design
- SQL and PL/SQL
- **Oracle Text**
- Oracle and XML
- Apex
- Object-Relational Extensions

Contents

- Foundations of Information Retrieval
- Oracle Text – overview
- Store, filter and index documents
- Query documents
- Extended features (Markup, Snippets etc.)
- New Features in Oracle 12*c*

NOT in THIS presentation:

- Administration
- Oracle Text and XML
- Oracle Text and JSON

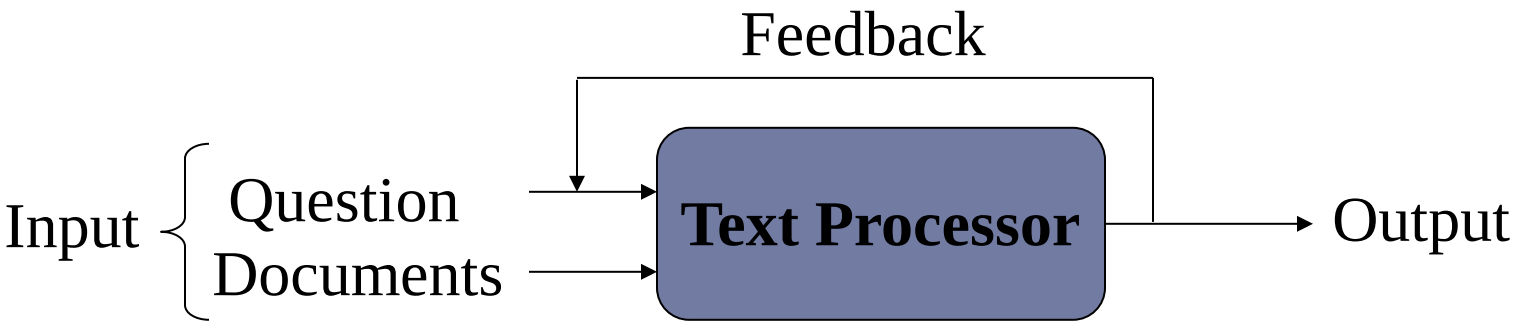
The challenge

80 % of enterprise data are **un-structured**. This means that they are inside documents of different kind, e.g. Word, PDF, Powerpoint. Catchphrases are:

- Document Workflow
- Content Management
- Knowledge Management

This requires intelligent **Text Retrieval**, which gives access to the **contents** of large collections of thematically complex documents written in **natural language**.

Task of Text Retrieval systems



Typical Tasks (Tracks of TREC):

- Ad hoc queries
- Known item search

  } Hit list of documents

- Filtering — Accumulated mass of relevant documents
- Novelty — Search for new nonredundant information
- Question Answering — Answer to a question, no retrieval of documents
- Relevance Feedback

TREC = **T**ext **RE**trieval **C**onference, organized annually by NIST since 1992

Correctness vs. Relevance

A **query on relational data** should have a correct result. The question

```
SELECT salary FROM employees WHERE name = 'Scott';
```

gives the salary of all employees named „Scott“. If I‘m interested in the salary of a particular „Scott“, then I must modify the question.

The **text query**

```
SELECT salary FROM employees WHERE CONTAINS(resume, 'Scott')>0;
```

may return also the salary of employees where, e.g., the supervisor‘s or teacher‘s name is „Scott“ .

Data vs. Information Retrieval

	Data Retrieval	Information Retrieval
Matching	exact	partial, best
Inference	deduction	induction
Model	deterministic	probabilistic
Classification	monothetic	polythetic
Query language	artificial	natural
Query specification	complete	incomplete
Items wanted	matching	relevant
Error response	sensitive	insensitive

See: C. J. van Rijsbergen: Information Retrieval, 1975

http://www.dcs.gla.ac.uk/Keith/Preface.html

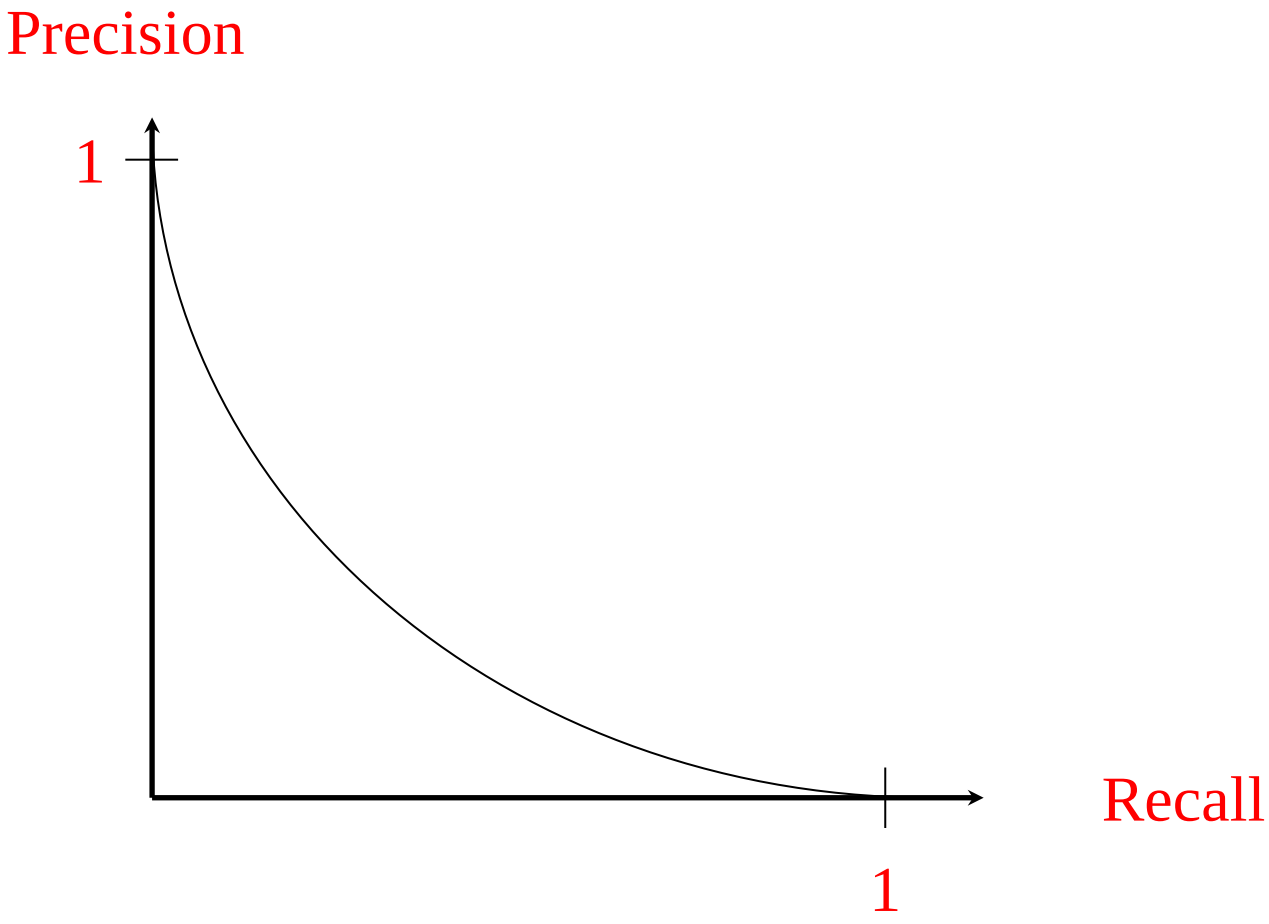
Evaluation of the results

Precision	Ratio of the returned relevant documents to the total number of returned documents
Recall	Ratio of the returned relevant documents to the number of relevant documents in the collection

Problems:

- Evaluation of the relevance
- Weighting the questions
- Position of the document in the ranking („Precision at n“)

Ratio between Precision and Recall



Documents in the database

	Storage	Analysis
Level 1	File system	Proprietary external applications
Level 2	Unstructured inside the database (VARCHAR2, LOB)	SQL Oracle Text
Level 3	Semistructured inside the database (XMLType)	SQL/XML, XQuery Oracle XML DB

Text Retrieval with the database

- Only one platform (Oracle DB)
- The manipulation of un-structured data is under the concurrency control of the database
- Synchronous storage of documents and metadata in columns of relational tables
- Unstructured data (text) can be analyzed using SQL
- Aggregation and Data (Text) Mining are available for un-structured data

Oracle Text – history

- Oracle 7: **NEW** additional software with separate license.
- Oracle 8*i*: element of **Oracle Intermedia** (extra cost option)
- Since Oracle 9*i*: **Integral part of all database editions**, including the Express Edition. **No extra cost.**
- Enhancements and new features in all subsequent releases including 12*c*

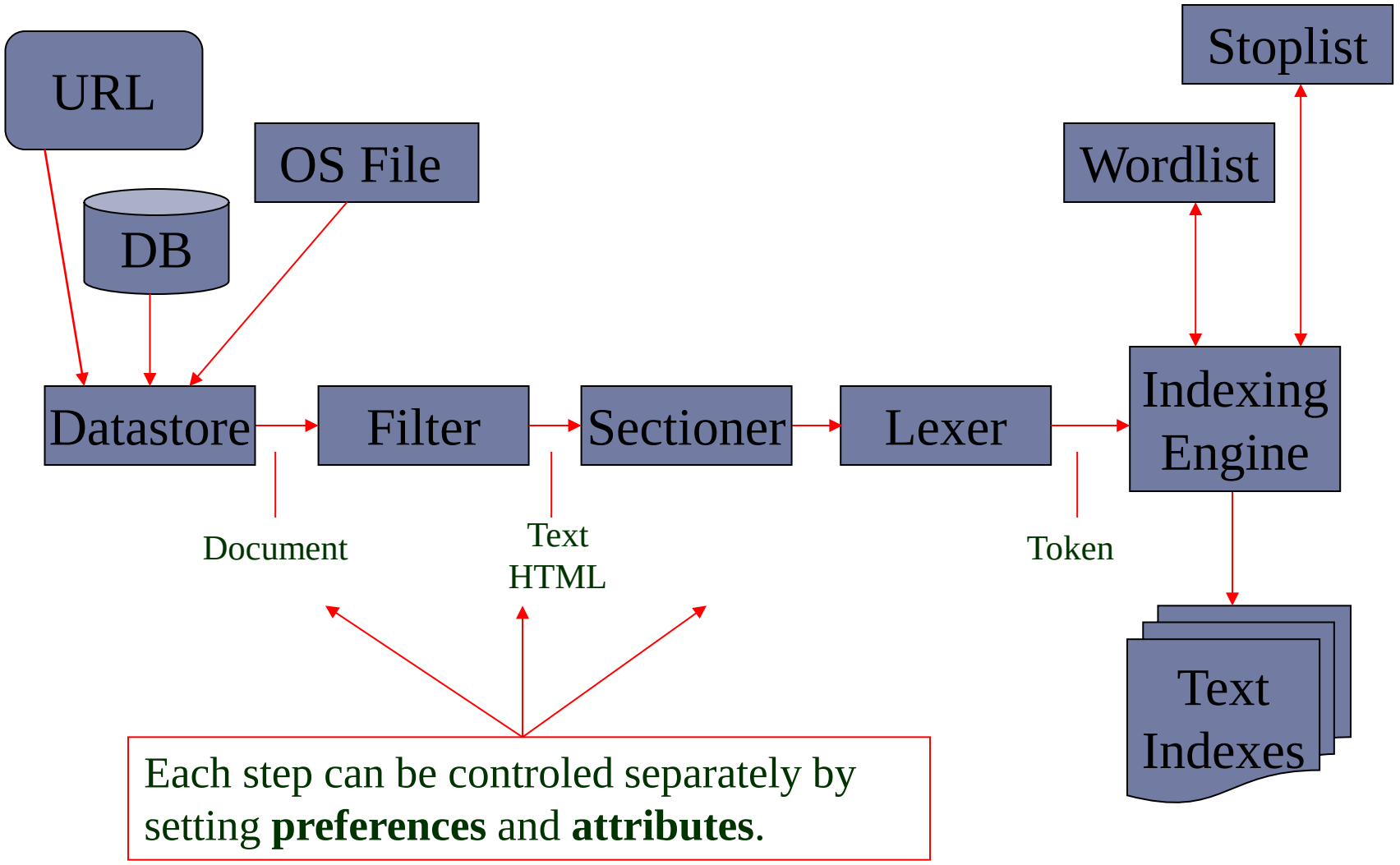
Oracle Text is based on **Data Cartridge** (Oracle‘s enhancement API)

Oracle Text‘s logic is executed server-side based on specific SQL and PL/SQL code.

Oracle Text is less in the centre of public interest as a top feature:

- No Oracle University classroom training is available
- No text book except the documentation

Indexing – core of Oracle Text



Types of text indexes

1. **CONTEXT**: For big collections of documents with different origin (e.g., externally: Word, HTML, XML, PDF; internally: BLOB, CLOB, VARCHAR2, BFILE) ➡ **CONTAINS**
2. **CTXCAT**: For short texts and mixed queries (text and metadata), reduced availability of query operators ➡ **CATSEARCH**
3. **CTXRULE**: Classification and routing of documents, index is on a set of query strings, not on the documents ➡ **MATCHES**
4. CTXXPATH: For XMLType columns, accelerates XML queries (de-supported in 12c) ➡ ExistsNode

CONTEXT index – syntax

```
CREATE INDEX text_idx ON schema.table(column)
INDEXTYPE IS CTXSYS.CONTEXT [ONLINE]
[PARAMETERS('paramstring')];
```

Parameter (examples):

- DATASTORE: DIRECT_DATASTORE, FILE_DATASTORE
- FILTER: NULL_FILTER, AUTO_FILTER
- LEXER: BASIC_LEXER, MULTI_LEXER, WORLD_LEXER

CONTEXT index – attributes (examples)

FILE_DATASTORE	path	string
BASIC_LEXER	continuation	character (e.g. - \)
	punctuations	character (. ,)
	whitespace	character
	newline	NEWLINE, CARRIAGE_RETURN
	skipjoin	character (e.g. -)
	printjoin	character (e.g. _)
	base_letter	YES/NO
	mixed_case	YES/NO
	composite	DEFAULT, GERMAN, DUTCH
	alternate_spelling	NONE, GERMAN, DANISH, SWEDISH
	new_german_spelling	YES/NO
	index_themes	YES/NO
BASIC_WORDLIST	stemmer	AUTO, NULL, ENGLISH, GERMAN
	fuzzy_match	GENERIC, AUTO, ENGLISH, GERMAN
	substring_index	FALSE/TRUE
	prefix_index	FALSE/TRUE
	wildcard_maxterms	20000

CONTEXT index – queries

```
SELECT id, SCORE(1) FROM table WHERE
CONTAINS(column, 'query_string',1)>0;
```

Examples for query strings:

'Franken ~ Nürnberg'	NOT Operator
'DB = database'	EQUIValence Operator
'Oracle, text, preference*3'	ACCUMulate Operator

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

f = frequency of the term in the document
N = number of documents
n = number of documents that contain the term

Result:

ID	SCORE(1)
2	70
3	35
1	4

The query scans only the index but not the document!

Attention: The value of SCORE is not the frequency of the term, but a ratio, given by *Saltons formula,* running from 1 to 100: For a high score a term must be frequent in the document but rare in the collection. (Inverse frequency scoring) Commonly occurence of the term in all documents is evaluated as noise by the Text Engine.

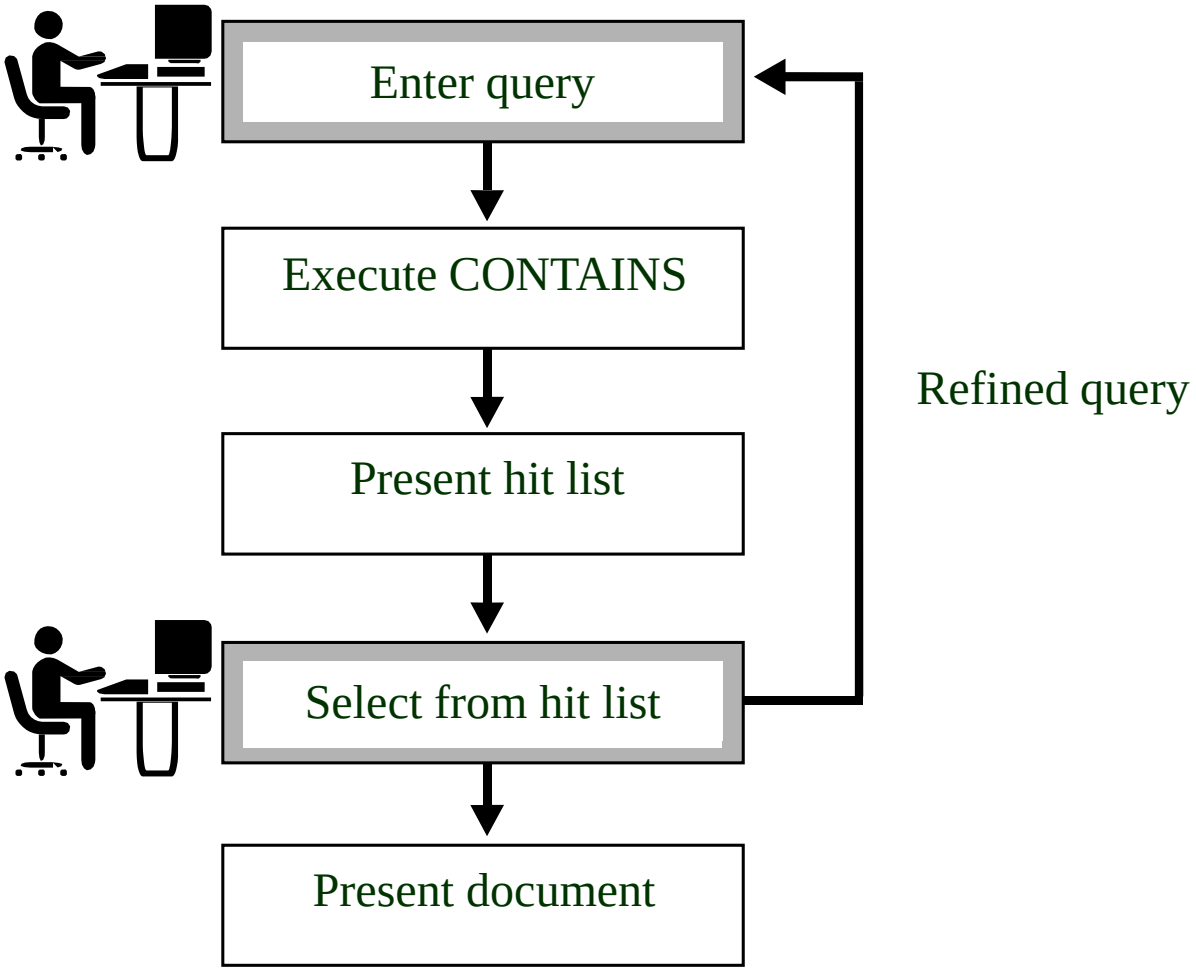
Query operators (examples)

Operator	Function
FUZZY	Similar words (Meier, Mayer, Meuer)
NEAR (;)	Two terms at a distance
stem ($)	Terms with the same root
WITHIN	Search a structure (paragraph, heading etc.)
INPATH	Search XML documents by path
SQE	Stored Query Expression (in dictionary)
Narrower Term (NT)	Search topic and subtopic *
ABOUT	Thematic search **

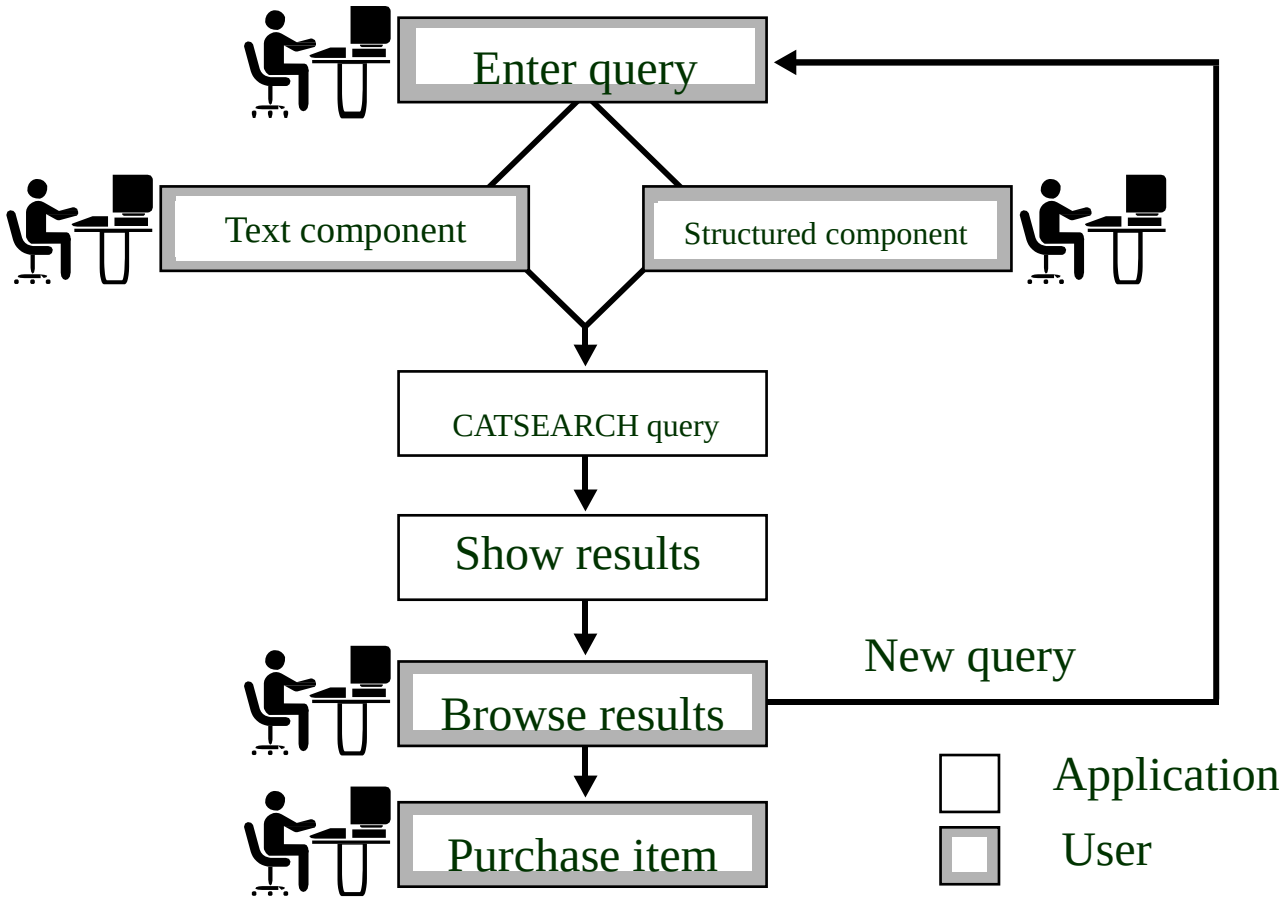
* Requires a registered **Thesaurus**

** Requires the compilation of the **Thesaurus** to a **Knowledge Base**

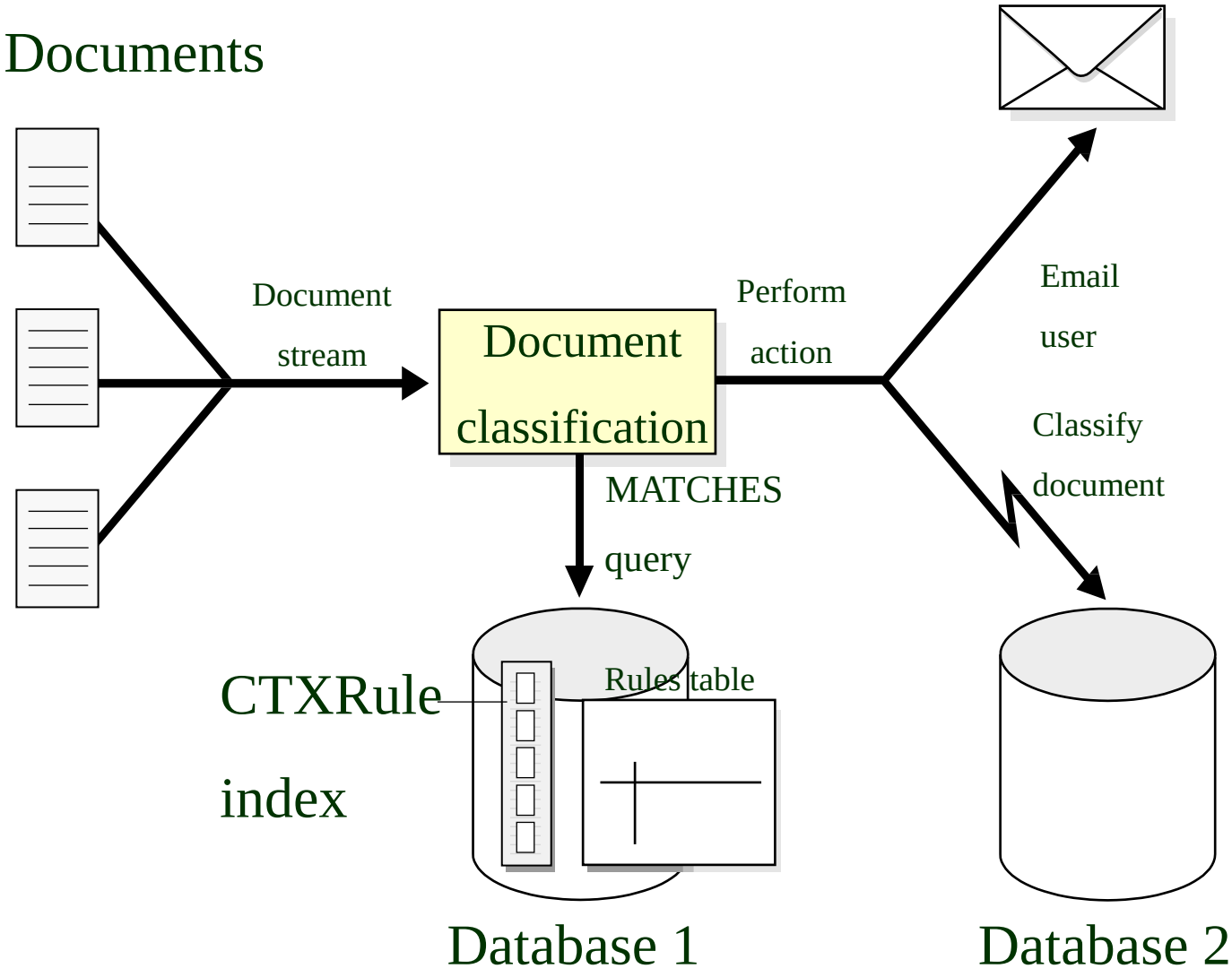
Text query – CONTAINS



Catalog query – CATSEARCH



Document classification (routing)



Extended features

Filtering	Filtering of more than 200 Formats to HTML or Text [1]
Highlighting	OFFSET and LENGTH of the founded terms
Markup	Marking of the founded terms
Navigation	Adding of navigation tags around the marked terms
Snippets	Output of the terms in their context
Gist	Thematic summary of a document [2]
Themes	Output of all topics of a document [2]

1) Including all usual word processing, desktop publishing, spread sheet, presentation and archiving and mail formats.
2) Requires a compiled Knowledge Base

Example Markup and Navigation

```
begin
   ctx_doc.markup(index_name => 'TEXT_IND_1',
   textkey => '1',
   text_query => 'history',
   restab => 'MARKUP_TEST',
   query_id => '1',
   tagset => 'HTML_NAVIGATE');
end;
```

The < **history** > of text retrieval is almost as long as the < **history** > of the written word. We know that Pliny the Elder (died 79 A.D.) used a Table of Contents structure to help readers navigate his mammoth "The Natural < **History** > in 37 Books". In this work, he refers to Valerius Soranus who, in the second century BC, was the first in Latin literature to use a Table of Contents. And in the third century B.C., Kallimachos of Kyrene created the first subject catalog to keep track of the scrolls at the Great Library of Alexandria in Egypt.

Snippets

Syntax

```
CTX_DOC.SNIPPET(
    index_name IN VARCHAR2,
    textkey IN VARCHAR2,
    text_query IN VARCHAR2,
    starttag IN VARCHAR2 DEFAULT '<b>',
    endtag IN VARCHAR2 DEFAULT '</b>',
    entity_translation IN BOOLEAN DEFAULT TRUE,
    separator IN VARCHAR2 DEFAULT '<b>...</b>'
)
return varchar2;
```

Example

```
select ctx_doc.snippet('CLOB_TAB_IDX', '1',
'microsoft near patch') from dual;
```

Result

```
SNIPPET
--------------------------------------------------------------------
refer to [NOTE:77627.1].
<b>Microsoft</b> Service Pack and Oracle <b>Patch</b> Set Support
================
```

Name search with NDATA section

Preparation

```
create table names (
  id number(10),
  name varchar2(200));
insert into names values (1, '<name>Max Mustermann</name>');
insert into names values (2, '<name>Larry Ellison</name>');
insert into names values (3, '<name>Ulrike Schwinn</name>');
insert into names values (4, '<name>Carsten Czarski</name>');
insert into names values (5, '<name>Günther Stürner</name>');

begin
  ctx_ddl.create_section_group('name_sg', 'BASIC_SECTION_GROUP');
  ctx_ddl.add_ndata_section('name_sg', 'name', 'name');
end;
/


create index ft_names on names (name)
indextype is ctxsys.context
parameters ('section group name_sg');
```

Result

```
select * from names where contains(name,'NDATA(name,Saarski Karsden)')>0;
 ID   NAME
----  ------------------------------
   4  <name>Carsten Czarski</name>
```

Mixed Query (Composite Domain Index)

Create index

```
CREATE INDEX comp_ind ON sh.customers(cust_first_name)
INDEXTYPE IS ctxsys.context
FILTER BY cust_id
ORDER BY cust_year_of_birth;
```

Query

```
SELECT /*+ first_rows(10) */ cust_id
FROM (select cust_id, cust_first_name,cust_year_of_birth from sh.customers
WHERE contains (cust_first_name, 'A% or D% or N% or B%',1)>0  AND cust_id>100000
ORDER BY cust_year_of_birth, score(1))
WHERE rownum<10;
```

Execution plan

```
| Id  | Operation                     | Name      | Rows  | Bytes | Cost (%CPU)
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |           |     1 |    13 |     5  (20)
|*  1 |  COUNT STOPKEY                |           |       |       |
|   2 |   VIEW                        |           |     1 |    13 |     5  (20)
|*  3 |    SORT ORDER BY STOPKEY      |           |     1 |    25 |     5  (20)
|   4 |     TABLE ACCESS BY INDEX ROWID| CUSTOMERS |     1 |    25 |     4   (0)
|*  5 |      DOMAIN INDEX             | COMP_IND  |       |       |     4   (0)
-------------------------------------------------------------------------------

5 - access("CTXSYS"."CONTAINS"("CUST_FIRST_NAME",'A% or D% or N% or B%',1)>0)
    filter("CUST_ID">100000)
```

New in Oracle 12c

- Performance
  - BIG_IO:TOKEN_INFO is stored in one big Secure File and not in inline BLOBs
  - SEPARATE_OFFSET: Offsets are stored separately
- Native Support for Snippets in Result Set Interface
- Forward Index: Documents are stored compressed within index for faster Highlighting and Snippeting
- SAVE COPY
- XQuery fulltext
- Automatic Near Real-Time Indexing with Staging Index
- Language recognition with POLICY_LANGUAGE
- Document-Level Lexer
- Definition of stop classes with regular expressions
- Enhancement of the query syntax
- SDATA and MDATA enhancements

Forward Index

- Highlighting and Snippets work in the following way:
  - Positions of the items are stored in table $I.
  - A document with higlighted search items is generated during runtime.
  - This may decrease the performance in case of big documents.
- Forward Indexing means:
  - An additional table $O exists for the offsets. It represents a mapping to the $I.
  - Higlighting, Snippet and Markup are performed on its base.
  - ```
    exec ctx_ddl.set_attribute
    ('mystore','forward_index','TRUE');
    ```
    ```

Save Copy

- Additional to Forward Index
  - Entire documents are stored in table $D
  - PLAINTEXT for SNIPPET
  - FILTERED for MARKUP or HIGHLIGHTING
  - NONE for VARCHAR2 or CLOB columns
  - ```
    exec ctx_ddl.set_attribute
    ('mystore','save_copy','PLAINTEXT');
    ```
- The contents of an additional column decide if the document is stored in $D:
  - ```
    create table docs( id number, txt varchar2(64),
    save varchar2(3) );
    ```
  - ```
    insert into docs values(1, 'hello world',
    'PLAINTEXT');
    ```
  - ```
    create index idx on docs(txt) indextype is
    ctxsys.context
    parameters('save_copy column save');
    ```

Near Real-time Indexing

- DML on the text column requires index maintenance
  - Synchronization: new entries will be added at the end
  - Optimization: new entries will be integrated
- Heavy DML: Index may be steal or the performance will decrease
- Alternative: Option STAGE_ITAB
  - New entries are written into the staging table $G. On $G is a B*Tree index $H (analog to $I and $X)
  - The MERGE method of the optimization integrates the entries `optlevel => CTX_DDL.OPTLEVEL_MERGE`
  - Table $G are kept in KEEP pool of the buffer cache
  - `exec ctx_ddl.set_attribute('mystore', 'STAGE_ITAB', 'YES');`
  - `g_table_clause` and `g_index_clause` include storage options

More information

- Further reading:
  - Oracle Text Reference
  - Oracle Text Application‘s Developer Guide
  - http://www.oracle.com/technetwork/database/enterprise-edition/index-098492.html
  - http://oracle-text-de.blogspot.com/
  - http://trec.nist.gov/
  - C.J. van Rijsbergen: Information Retrieval, Glasgow 1979
  - G. Salton, M. McGill: Introduction to Modern Information Retrieval, McGraw-Hill 1983
- Classroom training
  - http://training.ordix.de/siteengine/action/load/kategorie/Oracle/nr/1824/index.html
  - Next: 30.11. – 02.12.2015, 21.03. – 23.03.2016 …

Thank you!

Q & A

Dr. Frank Haney
info@haney.it
Tel.: 03641-210224

ORACLE | CERTIFIED PROFESSIONAL