Текстовые документы в Oracle: разнообразие источников, форматов, запросов

Владимир Пржиялковский
Преподаватель технологий Oracle
prz@yandex.ru
http://www.ccas.ru/prz

Вы думаете, мне это легко далось? Я работал над источниками.

И. Ильф, Е. Петров. Золотой теленок. РАССКАЗ БУХГАЛТЕРА БЕРЛАГИ.

Другие источники документов

Пример таблицы, рассмотренной в предыдущей статье, не вполне реалистичен, так как размер документов в нем ограничивался максимум четырьмя тысячами байтов для типа VARCHAR2. В то же время Oracle позволяет создавать индекс типа CTXSYS.CONTEXT еще на поля типа CLOB, XMLTYPE и даже BFILE и URITYPE. Выполним:

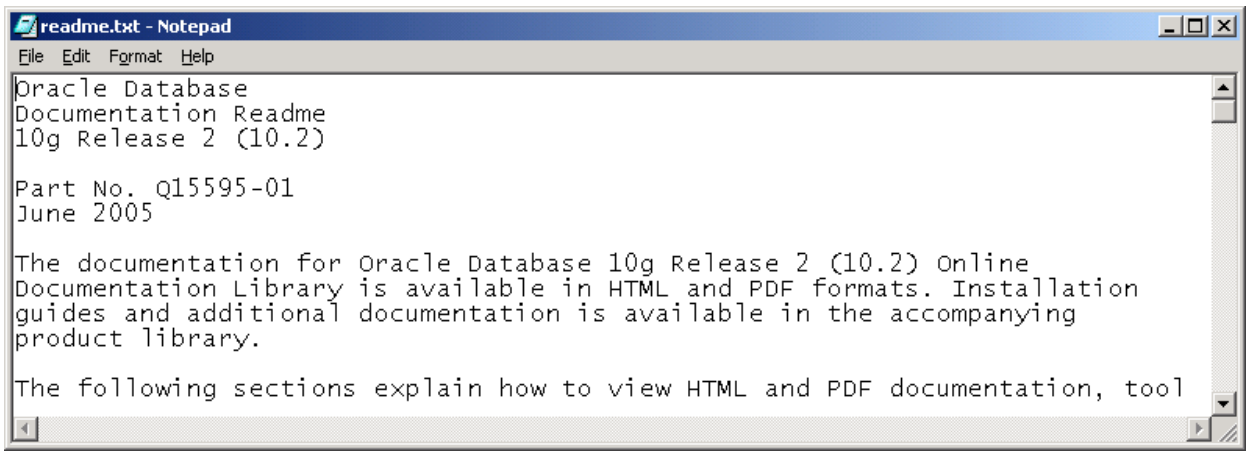
```
TRUNCATE TABLE docs;
DROP INDEX docs_vc2doc_idx;
ALTER TABLE docs DROP COLUMN vc2doc;
ALTER TABLE docs ADD ( clobdoc CLOB );

INSERT INTO docs VALUES ( 1, 'Mary had a little lamb' );
INSERT INTO docs VALUES ( 2, 'Twinkle, twinkle little star' );
INSERT INTO docs VALUES ( 3, 'This Lamb is my lamb' );

CREATE INDEX docs_clobdoc_idx ON docs ( clobdoc ) INDEXTYPE IS
ctxsys.context;
```

Проверка:

```
CTX> SELECT CONTAINS ( clobdoc, 'little' ) AS score FROM docs;

     SCORE
----------
         4
         4
         0
```

Следующий пример показывает, что Oracle позволяет создавать в БД текстовый индекс на документы, находящиеся вне базы.

Пусть на сервере имеется каталог *c:\distr\ora102\docdisk* с документацией по Oracle. Там есть простой текстовый файл *readme.txt*:



Создадим в БД указатель на каталог и переопределим таблицу DOCS:

```
CONNECT / AS SYSDBA
CREATE OR REPLACE DIRECTORY docs_dir AS 'c:\distr\ora102\docdisk';
GRANT READ ON DIRECTORY docs_dir TO ctx;
CONNECT ctx/ctx
TRUNCATE TABLE docs;
DROP INDEX docs_clobdoc_idx;
ALTER TABLE docs DROP COLUMN clobdoc;
ALTER TABLE docs ADD ( bfiledoc BFILE );

INSERT INTO docs VALUES ( 1, BFILENAME ( 'DOCS_DIR', 'readme.txt' ) );

CREATE INDEX docs_bfiledoc_idx ON docs ( bfiledoc ) INDEXTYPE IS
ctxsys.context;
```

Проверка:

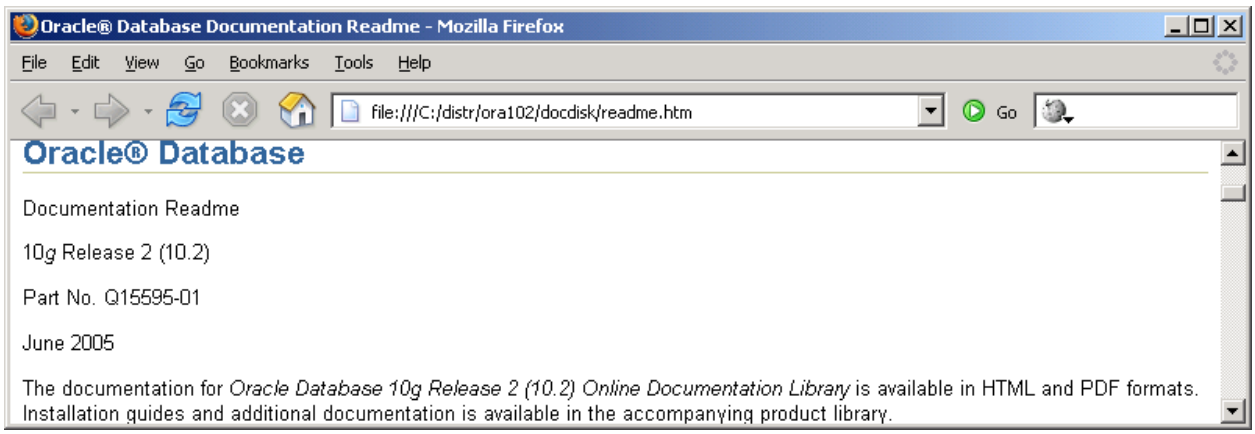
```
CTX> SELECT CONTAINS ( bfiledoc, 'oracle support' ) AS score FROM docs;

     SCORE
----------
        12
```

Обратите внимание, что в отличие от предыдущих примеров здесь документы хранятся в файловой системе, а в БД создается текстовый индекс; именно его и использует СУБД для вычисления результатов, несмотря на то, что формально запрос обращается к документам. Это может приводить к ошибкам при попытке извлечь сам документ ввиду его исчезновения уже после создания индекса – картина вполне привычная для тех, кто пользуется поисковыми машинами в интернете.

Как и раньше, изменения в документах не отразятся в индексе сами собой. Однако при хранении документов в БД система имела возможность фиксировать факт их изменения и предоставляла информацию о рассогласовании содержимого индекса и документов, чем можно было пользоваться, решая, стоит ли обновить индекс; в случае же внешнего хранения документов сведения о возможных рассогласованиях накапливаться в БД не могут.

Другие форматы документов

В том же каталоге файловой системы есть версии содержимого *readme.txt* в других форматах: это *readme.htm* и *readme.pdf*. Файл формата HTML имеет следующий вид:



Выполним:

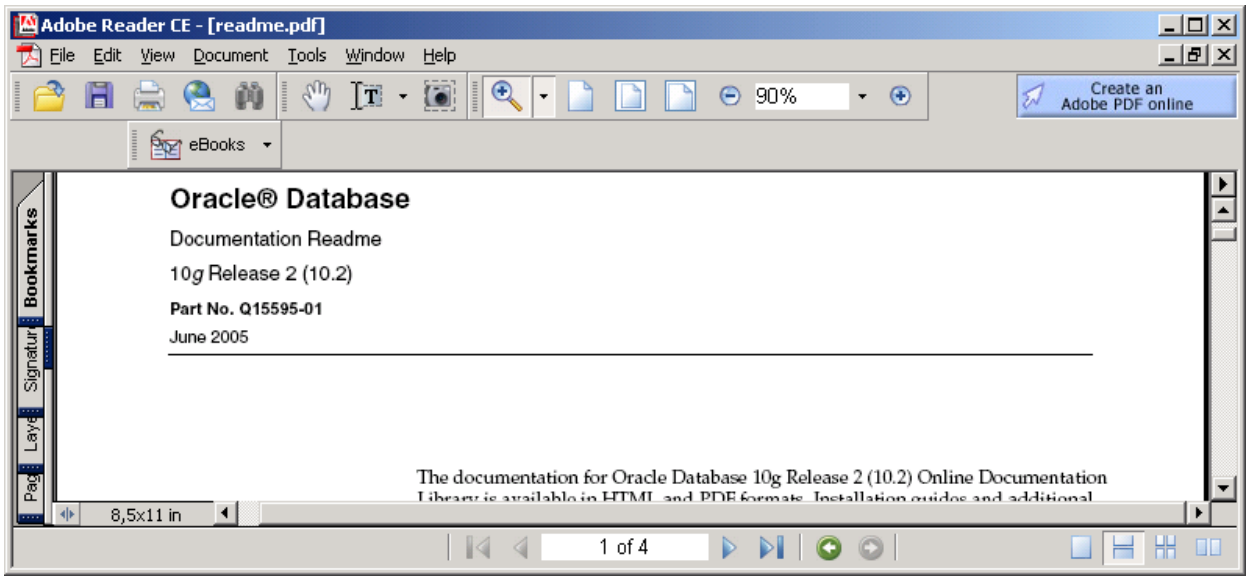
```
TRUNCATE TABLE docs;
DROP INDEX docs_bfiledoc_idx;
ALTER TABLE docs DROP COLUMN bfiledoc;
ALTER TABLE docs ADD ( htmldoc BFILE );

INSERT INTO docs VALUES ( 1, BFILENAME ( 'DOCS_DIR', 'readme.htm' ) );

CREATE INDEX docs_htmldoc_idx ON docs ( htmldoc )
INDEXTYPE IS CTXSYS.CONTEXT
PARAMETERS (
   'filter CTXSYS.NULL_FILTER section group CTXSYS.HTML_SECTION_GROUP'
);
```

В последней команде потребовалось нарушить предшествовавшую практику использования умолчаний и открыто указать в определении текстового индекса некоторые его параметры.

Проверка:

```
CTX> SELECT CONTAINS ( htmldoc, 'oracle support' ) AS score FROM docs;

     SCORE
----------
        12
```

Файл формата PDF имеет следующий вид:



Выполним:

```
TRUNCATE TABLE docs;
DROP INDEX docs_htmldoc_idx;
ALTER TABLE docs DROP COLUMN htmldoc;
ALTER TABLE docs ADD ( autodoc BFILE );

INSERT INTO docs VALUES ( 1, BFILENAME ( 'DOCS_DIR', 'readme.pdf' ) );

CREATE INDEX docs_autodoc_idx ON docs ( autodoc )
INDEXTYPE IS CTXSYS.CONTEXT
PARAMETERS (
   'filter CTXSYS.AUTO_FILTER section group CTXSYS.AUTO_SECTION_GROUP'
);
```

Вместо CTXSYS.**AUTO**_FILTER в параметрах индекса можно указать CTXSYS.**INSO**_FILTER. До версии 10 только так и нужно было поступать, однако с версии 10 фирма советует использовать новый AUTO-фильтр как более современную и совершенную реализацию старого INSO-фильтра (купленного в свое время фирмой Oracle у фирмы Inso). Фильтр используется СУБД для предварительной обработки текста перед построением индекса.

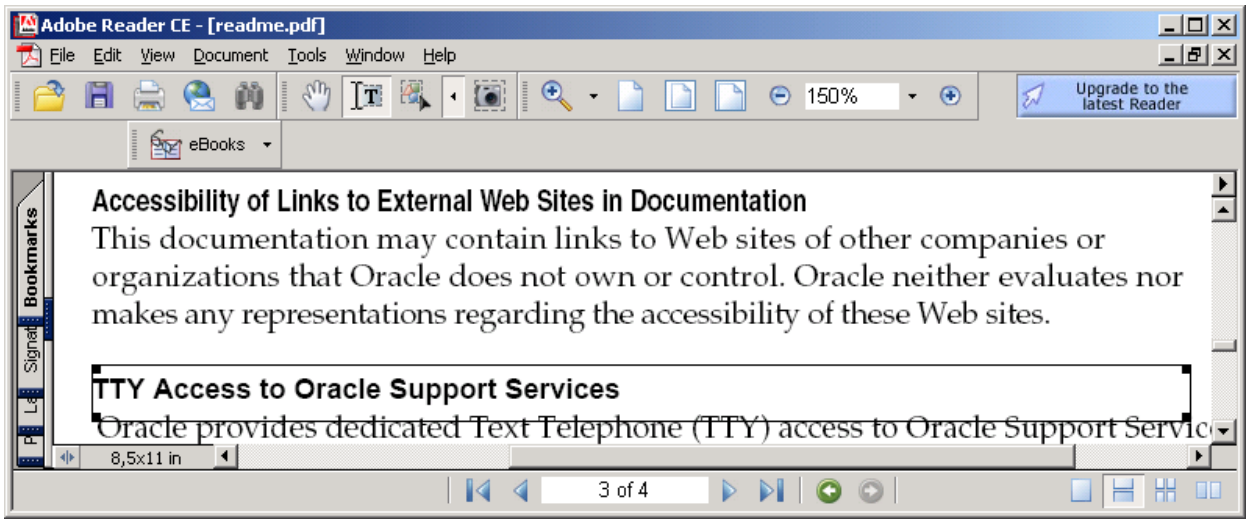
Проверка:

```
CTX> SELECT CONTAINS ( autodoc, 'oracle support' ) AS score FROM docs;

     SCORE
----------
         6
```

Обратите внимание на отличный от предыдущих примеров показатель соответствия документа запрашиваемой комбинации слов (6 против 12). Ручная проверка показывает, что сочетание 'oracle support' в каждом из текстов встречается одинаковое число раз, четырежды, так что степень соответствия всех документов должна быть одинакова. Последний результат является следствием особенности обработки документов PDF фильтром CTXSYS.AUTO_FILTER (до версии 10 CTXSYS.INSO_FILTER), примененном в построении индекса, и особенностями конкретного документа. В частности, согласно документации Oracle по версии 10, фильтр CTXSYS.AUTO_FILTER не замечает или «не обязательно правильно» обрабатывает:

- адреса в сети и электронной почты
- встроенные в документ шрифты
- версии PDF вне диапазона 1.1 (Acrobat 2.0) – 1.5 (Acrobat 6.0) (это относится к версии Oracle 10).

В нашем документе использована версия PDF 1.4, однако сам документ составлен неоднородно, что приводит к игнорированию при построении индекса последнего абзаца документа и его заголовка, в которых имеется два вхождения комбинации 'oracle support' из общих четырех (на это напоминает и внешний вид последнего абзаца):



Если бы документ *readme.pdf* был составлен «правильно», показатель его соответствия нашему запросу также был бы 12.

Досадные шероховатости обработки документов PDF компенсируются универсальностью AUTO/INSO-фильтра. Это универсальный фильтр, способный обработать при индексации документов большой перечень разных форматов, в том числе (помимо PDF) простой текстовый, HTML, DOC, RTF и ряд прочих (общим количеством более полутора сотен). Например, выполним:

```
INSERT INTO docs VALUES ( 2, BFILENAME ( 'DOCS_DIR', 'readme.txt' ) );
INSERT INTO docs VALUES ( 3, BFILENAME ( 'DOCS_DIR', 'readme.htm' ) );

EXECUTE CTX_DDL.SYNC_INDEX ( 'docs_autodoc_idx' )
```

Проверка:

```
CTX> SELECT CONTAINS ( autodoc, 'oracle support' ) AS score FROM docs;

     SCORE
----------
         6
        12
        12
```

В порядке *упражнения* предлагается проверить работу фильтра AUTO/INSO на файлах форматов DOC и RTF.

Конкретный формат документа фильтр AUTO распознает автоматически. Тем не менее, для некоторых популярных форматов фирма Oracle ради лучшей эффективности советует использовать специфичные фильтры: например, для формата HTML – тот, что был применен в примере выше. Фильтры (и прочие параметры текстового индекса) для форматов HTML и XML позволяют делать запросы с учетом разметки документов.

Параметры индекса

Параметры индекса позволяют задавать разные свойства индекса, например:

- фильтры для документа
- тип местонахождения документа
- тип лексического анализатора
- обеспечение индексом морфологического, нечеткого поиска; хранение префиксов
- учет структуры документа, такой как предложения, параграфы или разметка HTML/XML
- список неиндексируемых слов.

Иное название для параметров текстового индекса в Oracle – «предпочтения» (preferences).

Пример использования в качестве свойства индекса более удобного, чем в примерах выше, учета местонахождения документа:

```
TRUNCATE TABLE docs;
DROP INDEX docs_autodoc_idx;
ALTER TABLE docs DROP COLUMN autodoc;
ALTER TABLE docs ADD ( docname VARCHAR2 ( 100 ) );

INSERT INTO docs VALUES ( 1, 'c:\distr\ora102\docdisk\readme.txt' );
INSERT INTO docs VALUES ( 2, 'c:\distr\ora102\docdisk\readme.htm' );
INSERT INTO docs VALUES ( 3, 'c:\distr\ora102\docdisk\readme.pdf' );

CREATE INDEX docs_docname_idx ON docs ( docname )
INDEXTYPE IS CTXSYS.CONTEXT
PARAMETERS (
   'filter CTXSYS.AUTO_FILTER
    section group CTXSYS.AUTO_SECTION_GROUP
```

```
    datastore CTXSYS.FILE_DATASTORE'
);
```

Проверка:

```
CTX> COLUMN docname FORMAT A35
CTX> SELECT docname, CONTAINS ( docname, 'oracle support' ) FROM docs;

DOCNAME                             CONTAINS(DOCNAME,'ORACLESUPPORT')
----------------------------------- ---------------------------------
c:\distr\ora102\docdisk\readme.txt                                 12
c:\distr\ora102\docdisk\readme.htm                                 12
c:\distr\ora102\docdisk\readme.pdf                                  6
```

Более того, с каждым параметром связан один или более атрибутов. Однако явное указание атрибутов добавляет организационной сложности, так как производится техникой вызовов системных процедур, и не оформляется запросом SQL.

Поддержка текстовым индексом документов на русском

Приведенные выше примеры были для текстов на английском. Обработка текстов на разных языках имеет различия соответственно различиям устройства самих языков. Стандартная поставка Oracle Text способна работать со всеми языками, поддерживаемыми Oracle, но в рамках сравнительно простого контекстного поиска (о нем и шла речь выше), для которого различия языков несущественны. То есть контекстный поиск возможен и для документов на русском – это легко проверить в порядке *упражнения*. В этом отношении русский ничем не лучше эстонского или, скажем, языка телугу. Больше того, контекстный поиск в документах, по заверению документации Oracle, возможен не только для языков, перечисленных в таблице V $NLS_VALID_VALUES, но и для любого языка, кодировка которого включена в Unicode. Для этого, правда, требуется, чтобы Unicode была основной кодировкой для БД (это потребуется указать при создании базы).

В то же время продвинутый набор возможностей, включающий морфологический анализ, нечеткий поиск и другое, присутствует в готовом виде только для шести западноевропейских языков. Русского среди них нет.

Для тех, кому не повезло, разработчики Oracle Text дали механизм для самостоятельного построения продвинутых возможностей запросов. Мне известна всего одна попытка применить этот механизм к русскому языку, завершившаяся созданием готового продукта. Подробности можно найти по адресу *http://www.rco.ru/product.asp?ob_no=11*. Там же имеется на русском языке ряд материалов общего характера по обработке текстовых документов. Материалы вполне подтверждают догадку о том, что в конечном итоге сложность обработки текстовых документов сопоставима со сложностью естественных языков, то есть что работа с текстовыми документами в Oracle и где бы то ни было – это очень сложно.