

More Oracle Text Tips

NYOUG March 2002



Douglas Scherer, Core Paradigm



Agenda

- Quick Review
- Introduction to CTXCAT index type
- Using CTXCAT type indexes
- Introduction to Index Sets
- Using Index Sets
- Conclusion



Quick Review



Oracle Text Features

- Indexes any document or textual content to add fast, accurate retrieval of information to Internet content management applications, eBusiness catalogs, news services, job postings, etc.
- Adds powerful text search and intelligent text management to Oracle 9i
- Fully integrated with Oracle 9i
- Offers premier text search quality
- Contains several advanced features for text management, document services, and XML
- Has best internationalization set of features for multilingual text search applications

Excerpted from Oracle white paper, 2001



CONTEXT traits

- Rich set of document handling features
- Asynchronous coordination of index and table data
- Can make use of score value
- No index sets



Recipes table structure

```
SQL> DESC recipes
```

Name	Null?	Type
-----	-----	-----
ID	NOT NULL	NUMBER
NAME	NOT NULL	VARCHAR2 (100)
PREP_TIME_MINUTES		NUMBER
SERVINGS		NUMBER
DESCRIPTION		VARCHAR2 (1000)
COOKING_INSTRUCTIONS		CLOB
DISH_IMAGE		ORDSYS.ORDIMAGE
CULINARY_REVIEW		BLOB
CALS		NUMBER



Recipes table values

ID	NAME	CALS	SERVINGS
1	CB's Bean and Rice Soup	200	25
2	MC's tofu and rice surprise	100	10
3	Spanish Rice and Vegetable Stew	300	30



Using CONTEXT type index

- Create Index

```
CREATE INDEX recipes_name_ix  
ON recipes (name)  
INDEXTYPE IS CTXSYS.CONTEXT;
```

- Query

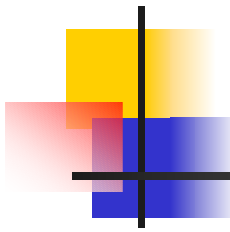
```
SELECT id, name  
FROM recipes  
WHERE CONTAINS(name, 'rice') > 0;
```



Query to find storage used by CONTEXT index

```
SELECT SUM(bytes)
  FROM user_segments
 WHERE segment_name IN
       (SELECT segment_name
        FROM user_lobs
        WHERE table_name LIKE 'DR$RECIPES_NAME_IX%'
        UNION ALL
        SELECT index_name
        FROM user_indexes
        WHERE table_name LIKE 'DR$RECIPES_NAME_IX%'
        UNION ALL
        SELECT table_name
        FROM user_tables
        WHERE table_name LIKE 'DR$RECIPES_NAME_IX%'
       );
```

Introduction to CTXCAT index type





CTXCAT traits

- Good with text fragments
- Index sets supporting mixed queries
- Transactional synchronization of index and table data
- No document handling features
- No score value
- Web-like operators



Creating CTXCAT index

- Create the Index

```
CREATE INDEX recipes_name_ix  
ON recipes (name)  
INDEXTYPE IS CTXSYS.CTXCAT;
```

- Query to find storage used by a CTXCAT index

```
SELECT SUM(bytes)  
FROM user_segments  
WHERE segment_name LIKE 'DR$RECIPES_NAME_IX%';
```



Using CTXCAT type indexes



CATSEARCH primer

- Operators in order of precedence
 - Grouping ()
 - Phrase " "
 - NOT -
 - AND
 - OR |
- CATSEARCH parameters
 - The name of the indexed column
 - The search string
 - The reference to one or more index sets.



CATSEARCH query examples:

Simple, OR, AND

- Simple

```
SELECT id, name
FROM recipes
WHERE CATSEARCH(name, 'rice', NULL) > 0;
```

- OR

```
SELECT id, name
FROM recipes
WHERE CATSEARCH(name, 'rice | bean', NULL) > 0;
```

- AND

```
SELECT id, name
FROM recipes
WHERE CATSEARCH(name, 'rice bean', NULL) > 0;
```



CATSEARCH query examples: NOT

- Correct use

```
SELECT id, name
  FROM recipes
 WHERE CATSEARCH(name, 'rice - bean', NULL) > 0;
```

```
SELECT id, name
  FROM recipes
 WHERE CATSEARCH(name, 'rice -bean', NULL) > 0;
```



CATSEARCH query examples: NOT (cont.)

- Illegal use

- Results in , "DRG-50901 : text query parser syntax error on line 1, column 1."

```
SELECT id, name
FROM recipes
WHERE CATSEARCH(name, '- rice - bean', NULL) > 0;
```

- Concatenation

- Interpreted as "ricebean"

```
SELECT id, name
FROM recipes
WHERE CATSEARCH(name, 'rice-bean', NULL) > 0;
```



CATSEARCH query examples: phrase, grouping

■ Phrase

```
SELECT id, name
  FROM recipes
 WHERE CATSEARCH(name, '"rice surprise"', NULL) > 0;
```

■ Grouping

```
SELECT id, name
  FROM recipes
 WHERE CATSEARCH(name, '(rice tofu) | spanish', NULL) > 0;
```



Introduction to index sets



Index set overview

- Index sets are used to support mixed queries
- Index sets hold indexes
 - each of those indexes is an ordered list of base table columns for use in mixed queries.
- Index sets are defined using the CTX_DDL package.
- Steps to create and implement an index set
 - 1. Create the index set
 - 2. Add indexes to the index set
 - 3. Create the CTXCAT type index specifying the index set(s)



Create the index set

- CTX_DDL.CREATE_INDEX_SET
 - SET_NAME VARCHAR2

```
SQL> EXEC CTX_DDL.CREATE_INDEX_SET('RECIPES_ISET')
```



Add indexes to the index set

- CTX_DDL.ADD_INDEX
 - SET_NAME (VARCHAR2)
 - COLUMN_LIST (VARCHAR2)

```
SQL> EXEC CTX_DDL.ADD_INDEX('RECIPES_ISET', 'CAL')
```

```
SQL> EXEC CTX_DDL.ADD_INDEX('RECIPES_ISET', 'SERVINGS')
```

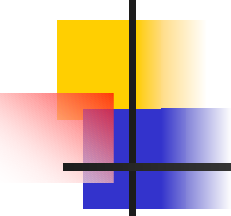


Create the CTXCAT type index specifying the index set(s)

```
CREATE INDEX recipes_name_ix  
ON recipes (name)  
INDEXTYPE IS CTXSYS.CTXCAT  
PARAMETERS ('index set recipes_iset');
```



Using index sets



Index set query example: ORDER BY

- This

```
SELECT id, name, calS
  FROM recipes
 WHERE CATSEARCH(name, 'rice',
                  'ORDER BY calS'
                  ) > 0;
```

- Versus

```
SELECT id, name, calS
  FROM recipes
 WHERE CATSEARCH(name, 'rice',
                  NULL
                  ) > 0

ORDER BY calS;
```



Execution plan comparison: ORDER BY

- Execution plan with use of index sets (This)

Execution Plan

```
0    SELECT STATEMENT Optimizer=CHOOSE
1 0    TABLE ACCESS (BY INDEX ROWID) OF 'RECIPES'
2 1      DOMAIN INDEX OF 'RECIPES_NAME_IX'
```

- Execution plan w/o use of index sets (Versus)

Execution Plan

```
0    SELECT STATEMENT Optimizer=CHOOSE
1 0    SORT (ORDER BY)
2 1      TABLE ACCESS (BY INDEX ROWID) OF 'RECIPES'
3 2        DOMAIN INDEX OF 'RECIPES_NAME_IX'
```



Index set query example:

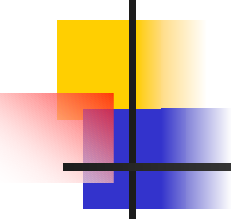
AND

- This

```
SELECT id, name,  
       FROM recipes  
       WHERE CATSEARCH(name, 'rice',  
                       'cals <= 100  
                       AND servings = 2'  
                       ) > 0;
```

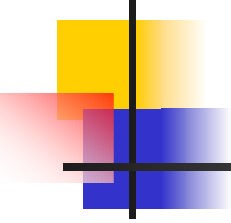
- Versus

```
SELECT id, name  
       FROM recipes  
       WHERE CATSEARCH(name, 'rice', NULL) > 0  
       AND cals <= 100  
       AND servings = 2;
```



Index set query example: complex (This)

```
SELECT id, name, cals,  
       servings  
FROM recipes  
WHERE CATSEARCH(name, 'rice',  
                'cals IN (100, 300)  
                AND servings = 2  
                ORDER BY servings'  
        ) > 0;
```



Index set query example: complex (Versus)

```
SELECT id, name, cals,  
       servings  
FROM recipes  
WHERE CATSEARCH(name, 'rice',  
                'cals IN (100, 300)  
                AND servings = 2'  
                ) > 0  
ORDER BY servings;
```



Index set rules

- An index set can take up to ninety-nine indexes
- NULLs are not allowed in a column used in an index set index. NULLs will cause an index error and the row will not be indexed.
- The only allowed data types are: NUMBER, DATE, CHAR, and VARCHAR2
- The maximum length of a column in an index set's index is thirty bytes.



Mixed query rules

- The left-hand side (the column name) of the expression must be a column named in at least one of the indexes of the index set.
- The left-hand side must be a column name.
- The operators are limited to: $<$, $<=$, $=$, $>=$, $>$, BETWEEN, and IN.
- The right-hand side must be composed of literal values.
- Criteria can be combined with AND
- All of the columns in an ORDER BY must go in the same direction.



Conclusion



CONTEXT/CTXCAT Comparison

■ CONTEXT

- Rich set of document handling features
- Asynchronous coordination of index and table data
- Can make use of score value
- No index sets

■ CTXCAT

- Better with text fragments
- Index sets supporting mixed queries
- Transactional synchronization of index and table data
- No document handling features
- No score value
- Web-like operators



Author Information

- Douglas Scherer
 - (dscherer@coreparadigm.com)
 - Douglas is president of Core Paradigm, a management consulting practice in New York. He is a frequent presenter at international conferences, author of books and articles on management and Oracle technology, and member of the adjunct instructional faculty at Columbia University's Executive Information Technology Management program.