



Tuning with Statistics

Wolfgang Breitling

breitliw@centrexcc.com



Agenda

The DBMS_STATS Package

Uses of DBMS_STATS beyond analyze

The “stattab” Table

Transferring Statistics with the stattab Table

Means to change an Access Plan

Using DBMS_STATS to change an Access Plan

DBMS_STATS

- ❖ Procedures for
- ❖ Gathering Statistics
- ❖ Getting or Setting Statistics
- ❖ Transferring Statistics



Gathering Statistics

GATHER_TABLE_STATS

GATHER_INDEX_STATS

GATHER_SCHEMA_STATS

GATHER_DATABASE_STATS

GATHER_SYSTEM_STATS

Oracle9i only

Current statistics in the specified table(s) can automatically be backed up before gathering new statistics.

GENERATE_STATS

Generates b-tree and bitmap index statistics from previously collected table statistics.

Getting or Setting Statistics

PREPARE_COLUMN_VALUES

CONVERT_RAW_VALUE

SET_COLUMN_STATS

DELETE_COLUMN_STATS

SET_INDEX_STATS

DELETE_INDEX_STATS

SET_TABLE_STATS

DELETE_TABLE_STATS

SET_SYSTEM_STATS

DELETE_SCHEMA_STATS

GET_COLUMN_STATS

DELETE_DATABASE_STATS

GET_INDEX_STATS

DELETE_SYSTEM_STATS

GET_TABLE_STATS

GET_SYSTEM_STATS



Transferring Statistics

CREATE_STAT_TABLE

DROP_STAT_TABLE

EXPORT_COLUMN_STATS

IMPORT_COLUMN_STATS

EXPORT_INDEX_STATS

IMPORT_INDEX_STATS

EXPORT_TABLE_STATS

IMPORT_TABLE_STATS

EXPORT_SCHEMA_STATS

IMPORT_SCHEMA_STATS

EXPORT_DATABASE_STATS

IMPORT_DATABASE_STATS

EXPORT_SYSTEM_STATS

IMPORT_SYSTEM_STATS

gather_table_stats

```
DBMS_STATS.GATHER_TABLE_STATS (  
  ownname,  
  tablename,  
  partname | NULL,  
  estimate_percent | NULL,  
  block_sample {TRUE | FALSE},  
  method_opt 'FOR ALL COLUMNS SIZE 1',  
  degree | NULL,  
  granularity {'DEFAULT | ALL | GLOBAL | PARTITION | SUBPARTITION'},  
  cascade {TRUE | FALSE},  
  no_invalidate {TRUE | FALSE},  
  stattab | NULL,  
  statid | NULL,  
  statown | NULL  
);
```

Oracle 9i

"gather_column_stats"

```
DBMS_STATS.GATHER_TABLE_STATS (
```

```
  ownname,
```

```
  tablename,
```

```
  method_opt 'FOR ALL [ INDEXED | HIDDEN ] COLUMNS SIZE n',
```

```
  method_opt 'FOR COLUMNS SIZE n col1 [, col2, ...]',
```

```
  method_opt 'FOR COLUMNS SIZE { repeat | auto | skewonly }'   Oracle9i  
);
```


gather_index_stats

```
DBMS_STATS.GATHER_INDEX_STATS (  
  ownname,  
  indname,  
  partname | NULL,  
  estimate_percent | NULL,  
  degree | NULL,  
  granularity {'DEFAULT | ALL | GLOBAL | PARTITION | SUBPARTITION'},  
  no_invalidate {TRUE | FALSE},  
  stattab | NULL,  
  statid | NULL,  
  statown | NULL  
);
```

Oracle 9i

gather_system_stats

Oracle 9i only

```
DBMS_STATS.GATHER_SYSTEM_STATS (  
    gathering_mode { NOWORKLOAD | INTERVAL | START | STOP },  
    interval n,  
    stattab | NULL,  
    statid | NULL,  
    statown | NULL  
);
```

create_stat_table

```
begin
  dbms_stats.create_stat_table (
    ownname =>          'scott',
    statab =>           'stats_table',
    tblspace =>         'tools');
end;
```

```
begin
  dbms_stats.drop_stat_table (
    ownname =>          'scott',
    statab =>           'stats_table');
end;
```

stat_table

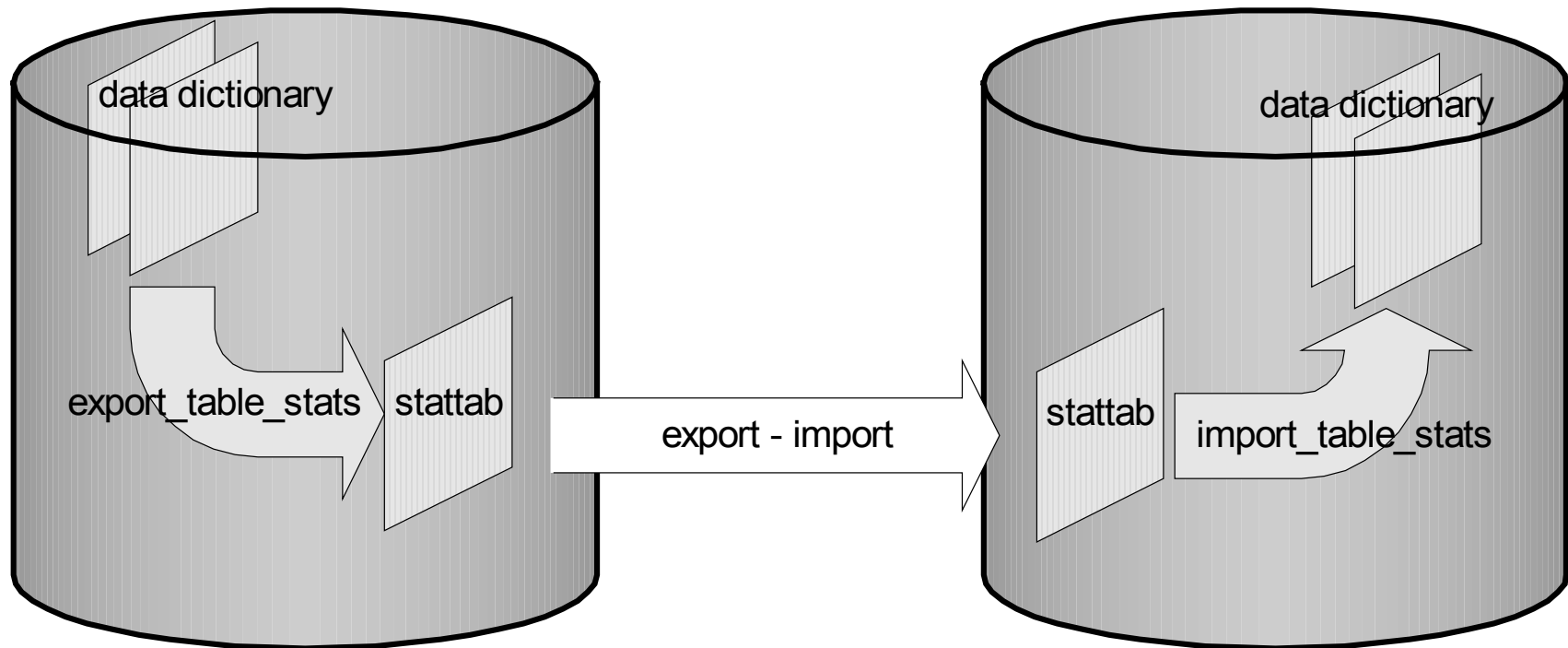
<u>Name</u>	<u>Null?</u>	<u>Type</u>	<u>Name</u>	<u>Null?</u>	<u>Type</u>
STATID		VARCHAR2 (30)	N1		NUMBER
TYPE		CHAR (1)	N2		NUMBER
VERSION		NUMBER	N3		NUMBER
FLAGS		NUMBER	N4		NUMBER
C1		VARCHAR2 (30)	N5		NUMBER
C2		VARCHAR2 (30)	N6		NUMBER
C3		VARCHAR2 (30)	N7		NUMBER
C4		VARCHAR2 (30)	N8		NUMBER
C5		VARCHAR2 (30)	N9		NUMBER
			N10		NUMBER
			N11		NUMBER
			N12		NUMBER
			D1		DATE
			R1		RAW (32)
			R2		RAW (32)
			CH1		VARCHAR2 (1000)

Uses for the stattab table

- ❖ Backup / Rollback Statistics Changes
- ❖ Transfer Statistics
 - ❖ Between databases
 - ❖ Within same database

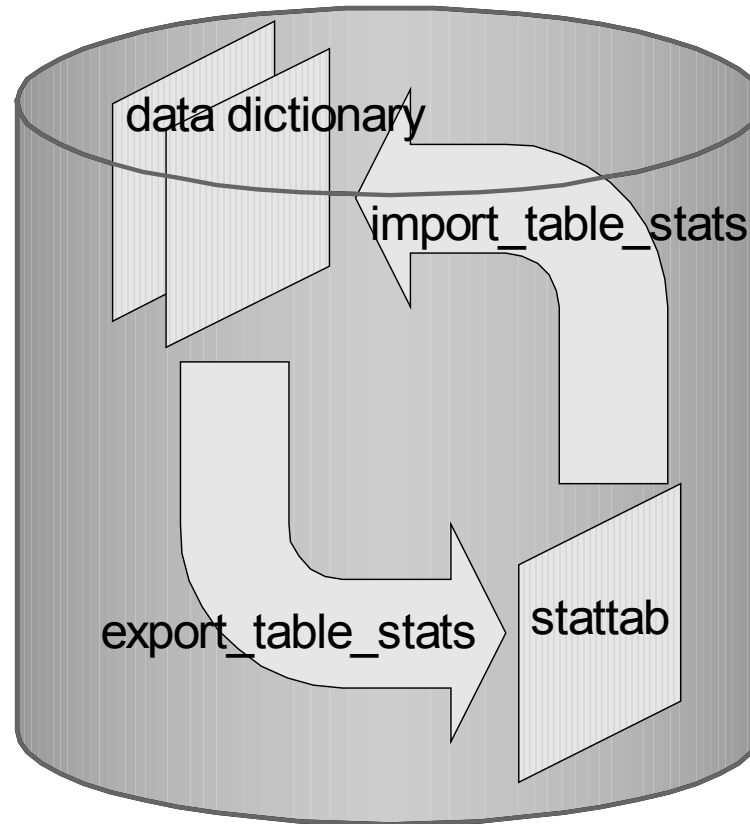
Transferring Statistics

between databases



Transferring Statistics

within a database



statown.stattab

Common columns

statid user settable identifier

type T table
C column
I index
S system

Oracle 9i

c5 owner null for system statistics

d1 last_analyzed null for system statistics

statown.stattab - type T

c1	table_name	n1	num_rows
c2	partition_name	n2	blocks
c3	subpartition_name	n3	avg_row_len
		n4	sample_size

statown.stattab - type C

c1	table_name	n1	num_distinct
c2	partition_name	n2	density
c3	subpartition_name	n4	sample_size
c4	column_name	n5	num_nulls
		n6	lo_value
		n7	hi_value
		n8	avg_col_len

statown.stattab - type C

c1	table_name	n9	? (always 1)
c2	partition_name	n10	endpoint_number
c3	subpartition_name	n11	endpoint_value
c4	column_name	n12	? (always null)

statown.stattab - type I

c1	index_name	n1	num_rows
c2	partition_name	n2	leaf_blocks
c3	subpartition_name	n3	distinct_keys
		n4	leaf_blocks_per_key
		n5	data_blocks_per_key
		n6	clustering_factor
		n7	blevel
		n8	sample_size

statown.stattab - type S

c4	'CPU_SERIO'	n1	sreadtim
c1	status	n2	mreadtim
c2	dstart	n3	cpuspeed
c3	dstop	n11	mbrc
c4	'PARIO'	n1	maxthr
		n2	slavethr



stattab views

stats_tables

stats_tab_partitions

stats_indexes

stats_ind_partitions

stats_tab_columns

stats_part_col_statistics

stats_tables view

```
create or replace view stats_tables (      as
      STATID      select STATID
      ,OWNER      ,C5
      ,TABLE_NAME ,C1
      ,NUM_ROWS   ,N1
      ,BLOCKS     ,N2
      ,AVG_ROW_LEN ,N3
      ,SAMPLE_SIZE ,N4
      ,LAST_ANALYZED) ,D1

from stats_table where type = 'T' and c2 is null
```

export_table_stats

```
export_table_stats (  
    ownname          varchar2,  
    tablename        varchar2,  
    partname         varchar2,  
    statid           varchar2,  
    statown          varchar2,  
    stattab          varchar2,  
    cascade          boolean  
);
```


Phases of SQL Tuning

- 1 Identify poorly performing SQL
- 2 Identify alternate – i.e. better – access plan
- 3 Persuade CBO to use this access plan

Means to change access plan

- ❖ Change the statement
- ❖ Use hint
- ❖ Change statistics
 - ❖ Create or drop an index
 - ❖ Change initialization parameters

Statistics affecting the CBO

Table

num_rows, blocks, avg_row_len

Column

num_distinct, density, num_nulls, buckets,
low_value, high_value

Index

blevel, leaf_blocks, distinct_keys,
avg_leaf_blocks_per_key,
avg_data_blocks_per_key, clustering_factor



Parameters affecting the CBO

optimizer_features_enable

optimizer_mode/goal

optimizer_search_limit

optimizer_max_permutations

optimizer_index_caching

optimizer_index_cost_adj

continued



Parameters affecting the CBO

db_file_multiblock_read_count

hash_area_size

hash_join_enabled

hash_multiblock_io_count

partition_view_enabled

sort_area_size

continued



Parameters affecting the CBO

star_transformation_enabled

optimizer_percent_parallel

parallel_broadcast_enabled

query_rewrite_enabled

query_rewrite_integrity

query_rewrite_expression

Backup, backup, backup

When manipulating the statistics, make two exports, one for backup, one to work with.

e.g.

```
DBMS_STATS.EXPORT_TABLE_STATS (  
  ownname => 'abc', tabname => 'xyz',  
  statab => 'stats_table', statid => 'bkup');
```

```
DBMS_STATS.EXPORT_TABLE_STATS (  
  ownname => 'abc', tabname => 'xyz',  
  statab => 'stats_table', statid => 'work');
```

Putting it to work – the SQL

```
SELECT A.ACCOUNT, SUM(A.POSTED_TOTAL_AMT)
from PS_PCR_LEDSUM_OP A
    , PSTREESELECT06 L1
    , PSTREESELECT10 L
where A.LEDGER = '*****'
    and A.FISCAL_YEAR = 1997
    and A.ACCOUNTING_PERIOD BETWEEN 1 and 12
    and A.CURRENCY_CD IN (' ', 'CAD')
    and A.STATISTICS_CODE = ' '
    and A.PCR_TREE_NODE_DEPT = '*****'
    and L1.SELECTOR_NUM = 15101
    and A.ACCOUNT = L1.RANGE_FROM_06
    and (L1.TREE_NODE_NUM BETWEEN 1968278454 and 1968301256
        OR L1.TREE_NODE_NUM BETWEEN 1968301263 and 1968301270
        OR L1.TREE_NODE_NUM BETWEEN 1968867729 and 196888696
        OR L1.TREE_NODE_NUM BETWEEN 1969156312 and 1969207615)
    and L.SELECTOR_NUM = 15109
    and A.DEPTID = L.RANGE_FROM_10
    and L.TREE_NODE_NUM BETWEEN 1692307684 and 1794871785
group by A.ACCOUNT
```


Putting it to work - before

<u>cost</u>	<u>card</u>	<u>operation</u>
130	1	SELECT STATEMENT
130	1	SORT GROUP BY
128	1	NESTED LOOPS
125	1	NESTED LOOPS
50	1	INDEX RANGE SCAN PSAPSTREESELECT06
75	1,753	TABLE ACCESS BY LOCAL INDEX ROWID PS_PCR_LEDSUM_OP PARTITION: START=5 STOP=5
2	1,753	INDEX RANGE SCAN PS_PCR_LEDSUM_OP_ACC PARTITION: START=5 STOP=5
3	456	INDEX RANGE SCAN PSBPSTREESELECT10

Putting it to work - before

<u>call</u>	<u>count</u>	<u>cpu</u>	<u>elapsed</u>	<u>disk</u>	<u>query</u>	<u>current</u>	<u>rows</u>
Parse	1	0.15	0.17	0	0	0	0
Exec	1	0	0	0	0	0	0
Fetch	4	2.03	19.54	15162	22909	0	34
total	6	2.18	19.71	15162	22909	0	34

<u>call</u>	<u>count</u>	<u>cpu</u>	<u>elapsed</u>	<u>disk</u>	<u>query</u>	<u>current</u>	<u>rows</u>
Parse	1	0	0	0	0	0	0
Exec	1	0	0	0	0	0	0
Fetch	4	1.19	1.21	12310	22909	0	34
total	6	1.19	1.21	12310	22909	0	34

<u>call</u>	<u>count</u>	<u>cpu</u>	<u>elapsed</u>	<u>disk</u>	<u>query</u>	<u>current</u>	<u>rows</u>
Parse	1	0	0	0	0	0	0
Exec	1	0	0	0	0	0	0
Fetch	4	1.14	1.19	11413	22909	0	34
total	6	1.14	1.19	11413	22909	0	34

Putting it to work - before

card	Rows	Execution Plan
1	0	SELECT STATEMENT GOAL: CHOOSE
1	34	SORT GROUP BY
1	892	NESTED LOOPS
①	1,125	NESTED LOOPS
1	151	INDEX RANGE SCAN PSAPSTREESELECT06
1,753	1274	TABLE ACCESS BY LOCAL INDEX ROWID PS_PCR_LEDSUM_OP PARTITION: START=5 STOP=5
1,753	31,538	INDEX RANGE SCAN PS_PCR_LEDSUM_OP_ACC PARTITION: START=5 STOP=5
456	892	INDEX RANGE SCAN PSBPSTREESELECT10

Join cardinality: 1 = outer (1) * inner (1753) * sel (7.0522e-04)

TABLE: PSTREESELECT06 ORIG CDN: 154506 CMPTD CDN: 1

Putting it to work

<u>table</u>	<u>column</u>	<u>NDV</u>	<u>density</u>	<u>bkts</u>
PSTREESELECT06	SELECTOR_NUM	158	5.8728E-02	5
PSTREESELECT06	TREE_NODE_NUM	2,619	5.5647E-03	5
PSTREESELECT06	RANGE_FROM_06	10,152	9.8503E-05	1
PSTREESELECT06	RANGE_TO_06	10,152	9.8503E-05	1

update stats_tab_columns set density = 10*density
where statid = 'WORK' and table_name = 'PSTREESELECT06'
and column_name = 'TREE_NODE_NUM';

<u>table</u>	<u>column</u>	<u>NDV</u>	<u>density</u>	<u>bkts</u>
PSTREESELECT06	SELECTOR_NUM	158	5.8728E-02	5
PSTREESELECT06	TREE_NODE_NUM	2,619	5.5647E-02	5
PSTREESELECT06	RANGE_FROM_06	10,152	9.8503E-05	1
PSTREESELECT06	RANGE_TO_06	10,152	9.8503E-05	1

Putting it to work - after

<u>cost</u>	<u>card</u>	<u>operation</u>
1,215	64	SELECT STATEMENT
1,215	64	SORT GROUP BY
1,209	64	HASH JOIN
4	273	INDEX RANGE SCAN PSAPSTREESELECT10
1,204	359	HASH JOIN
50	290	INDEX RANGE SCAN PSAPSTREESELECT06
1,153	1,753	TABLE ACCESS BY LOCAL INDEX ROWID PS_PCR_LEDSUM_OP PARTITION: START=5 STOP=5
59	1,753	INDEX RANGE SCAN PS_PCR_LEDSUM_OP_TDEP PARTITION: START=5 STOP=5

TABLE: PSTREESELECT06 ORIG CDN: 154506 CMPTD CDN: 290

Join cardinality: 359 = outer (290) * inner (1753) * sel (7.0522e-04)

Join cardinality: 64 = outer (359) * inner (273) * sel (6.5703e-04)

Putting it to work - after

<u>call</u>	<u>count</u>	<u>cpu</u>	<u>elapsed</u>	<u>disk</u>	<u>query</u>	<u>current</u>	<u>rows</u>
Parse	1	0.23	0.24	0	0	0	0
Exec	1	0	0	0	0	0	0
Fetch	4	0.35	0.51	248	655	0	34
total	6	0.58	0.75	248	655	0	34

<u>call</u>	<u>count</u>	<u>cpu</u>	<u>elapsed</u>	<u>disk</u>	<u>query</u>	<u>current</u>	<u>rows</u>
Parse	1	0	0	0	0	0	0
Exec	1	0	0	0	0	0	0
Fetch	4	0.18	0.19	29	655	0	34
total	6	0.18	0.19	29	655	0	34

<u>call</u>	<u>count</u>	<u>cpu</u>	<u>elapsed</u>	<u>disk</u>	<u>query</u>	<u>current</u>	<u>rows</u>
Parse	1	0	0	0	0	0	0
Exec	1	0	0.01	0	0	0	0
Fetch	4	0.19	0.19	0	655	0	34
total	6	0.19	0.2	0	655	0	34

Putting it to work - after

card	Rows	Execution Plan
64	0	SELECT STATEMENT GOAL: CHOOSE
64	34	SORT GROUP BY
64	892	HASH JOIN
273	150	INDEX RANGE SCAN PSAPSTREESELECT06
359	8752	HASH JOIN
290	9237	INDEX RANGE SCAN PSAPSTREESELECT10
1,753	10524	TABLE ACCESS BY LOCAL INDEX ROWID PS_PCR_LEDSUM_OP PARTITION: START=5 STOP=5
1,753	12842	INDEX RANGE SCAN PS_PCR_LEDSUM_OP_TDEP PARTITION: START=5 STOP=5



Metalink Notes

- 114671.1 Gathering Statistics for the Cost Based Optimizer
- 130899.1 How to Set User-Defined Statistics Instead of RDBMS Statistics
- 122009.1 How to Retrieve Statistics Generated by ANALYZE SQL Statement
- 130688.1 Report Statistics for a Table, it's columns and it's indexes with DBMS_STATS
- 130911.1 How to Determine if Dictionary Statistics are RDBMS-Generated or User-Defined
- 102334.1 How to automate ANALYZE TABLE when changes occur on tables
- 1074354.6 DBMS_STATS.CREATE_STAT_TABLE: What Do Table Columns Mean?
- 117203.1 How to Use DBMS_STATS to Move Statistics to a Different Database
- 149560.1 Collect and Display System Statistics (CPU and IO) for CBO usage
- 153761.1 Scaling the system to improve CBO optimizer



Resources

See also Dave Ensor's session paper for IOUG

asktom.oracle.com

(Thomas Kyte)

www.ixora.com.au

(Steve Adams)

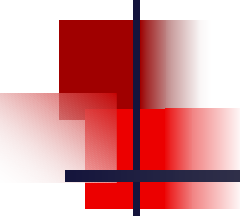
www.hotsos.com

(Cary Millsap)

www.orapub.com

(Craig Shallahamer)

www.jlcomp.demon.co.uk **(Jonathan Lewis)**



Wolfgang Breitling
breitliw@centrexcc.com
Centrex Consulting Corp.