

## 2.4.4 Runstats

Make sure Runstats accurately reflect table statistics. That doesn't mean to say Runstats has to be run every week. In fact, unless the table is volatile, it may not be necessary to run it more than once. Even with tables that are large, keep in mind that the Optimizer is unlikely to choose a different access path if the table has 100K rows in it or 1m rows. If it's not choosing a good index for 100K rows, it's not going to choose a better one for 1m rows. So once your environment is stable, consider running Runstats only once after an initial Reorg, and then don't run it again until any performance deterioration warrants it. It is probably better to gather more statistics against a table space less frequently, than less statistics more frequently. The more information the Optimizer has about a table, the more likely it is to choose the optimum access path. Keep in mind that the nature of dynamic SQL implies the access path can change any time the catalog statistics change. Erring on the side of "if it ain't broke, don't fix it" may sometimes be wise. If you need to use Runstats to track space usage or Reorg criteria, consider using Real-time statistics instead, described below.

There is an inline Runstats statement available with PeopleSoft. See the section on "%UPDATESTATS" and "Multi-instance temp tables" in Guide 3 for a discussion.

Some customers have reported better access paths by using the DB2 V5+ Runstats options of KEYCARD and FREQVAL to get better statistics on key columns. Instead of DB2 capturing distribution statistics for only the "FIRSTKEYCARD" (i.e. first column in the index), and "FULLKEYCARD" (i.e. all columns concatenated in the index), these keywords will capture distribution stats for each individual (or group of) column in the index. The more information the Optimizer can use, the more chance it has of getting the access path correct. There is also a utility available called DSTATS that will collect column frequency values on all columns if desired, and may help ad-hoc queries not using key columns. This can be found at the IDUG site ([www.idug.org](http://www.idug.org)) by searching on the keyword DSTATS.

The DSTATS utility is now incorporated into DB2 V8 allowing DB2 to gather the following additional statistics:

- Frequency value distributions for non-indexed columns or groups of columns
- Cardinality values for groups of non-indexed columns
- LEAST frequently occurring values, along with MOST for both index and non-indexed column distributions

This enhancement adds the new functionality of calculating the frequencies for non-indexed columns to RUNSTATS. The relevant catalog tables are updated with the specified number of highest frequencies and optionally with the specified number of lowest frequencies. The new functionality also optionally collects multi-column cardinality for non-indexed column groups and updates the catalog. When a query contains predicates on columns that are correlated and/or skewed, and the Optimizer does not have correlation and/or skew statistics, the Optimizer may incorrectly estimate the filtering for those predicate(s).

Version 8 relaxes the Version 7 requirement that history statistics can only be collected if the main catalog statistics (used by the Optimizer) are also updated. This greater flexibility allows you to keep track of statistics changes over time, without the concern that the main Optimizer statistics change as well and can result in access path changes, especially for dynamic SQL. So in DB2 V8, you can specify RUNSTATS UPDATE NONE HISTORY ALL. This way, all statistics are collected (ALL), but only the history tables are updated (UPDATE NONE).

Here's a reference chart that describes which columns get updated by which Runstats command.

SYSIBM.		1	2	3	4	5
<b>systablespace</b>						
	nactive	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<b>systables</b>						
	card	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	npages	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	pctpages	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	pctroscmp	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<b>syscolumns</b>						
	colcard		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	high2key		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	low2key		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<b>syscoldist</b>						
	frequency			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	name			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	colvalue			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	type			F	C	F
	cardf			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	numcolumns			1	n	n
<b>sysindexes</b>						
	clusterratio			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	firstkeycard			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	fullkeycard			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	nleaf			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	nlevels			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

- 1 Runstats tablespace(A)
  - Collects tablespace and all table statistics
- 2 Runstats tablespace (A) Table(T1)
  - Collects column cardinalities for a specific table (but still updates systables for all tables)
- Runstats tablespace(A) Table(ALL)
  - Collects column cardinalities for all columns of all tables
- 3 Runstats tablespace (A) table(ALL) index(I1)
  - Collects index statistics for a specific index plus frequencies for the leading index column
- Runstats tablespace (A) table(ALL) Index(ALL)
  - Collects index statistics for all indexes plus frequencies for the leading index column
- 4 Runstats tablespace (A) table(ALL) index(all keycard)
  - Collects cardinality statistics for each index key column combination (AB, ABC etc)
- 5 Runstats tablespace (A) table(ALL) index(all keycard freqval numcols 2 count 10 numcols 3 count 10)
  - Collects frequency statistics for each index key column combination specified (AB, ABC)

#### 2.4.4.1 Fast cached SQL statement invalidation

DB2 V5 APAR PQ17905 added support so that RUNSTATS on a table space or index would purge statements from the cache that referenced the table space or index. For people that are worried about updating catalog statistics and possibly changing access paths, they can run with the UPDATE (NONE) option. The cache will still be invalidated. In DB2 V8, this has been enhanced to also allow REPORT NO UPDATE NONE. There will be no data capture or computation cost.

## 2.4.5 Real-time statistics

Real-time statistics are a good way to determine when you need to perform certain maintenance operations on your table spaces without having to run Runstats. Many customers run Runstats on a regular basis to get space and "organization" data on their tables to determine when to run Reorgs. But as mentioned above, from purely an access path perspective, this may not be necessary. Real-time statistics provide a solution.

- DB2 will collect the statistics in real-time, in memory, and then periodically update TABLESPACESTATS and INDEXSPACESTATS
- The overhead of RTS is negligible compared to the added value of the information that is obtained.
- The counters in the RTS tables are always overwritten and DB2 does not provide a mechanism to keep track of history.
- The Optimizer does not use the real-time statistics tables. It only uses the "Runstats" tables.
- In general, Insert, Update, Delete statements cause DB2 to modify real-time statistics, along with certain utilities, including LOAD, REORG, REBUILD INDEX, RUNSTATS, COPY and RECOVER

Real time statistics are activated by

- Creating the database DSNRTSDB along with its tables and indexes.
- Setting the STATSINT Zparm to an appropriate value. The default is 1440
- Starting DSNRTSDB

The statistics are maintained in memory, but are externalized when any of the following occurs:

- Stop db (dsnrtsdb)
- Stop a table space (externalizes for only that table space)
- STATINT Time interval
- Stop db2 mode (quiesce)
- Utility operations