Database Statistics

Database Statistics provide an invaluable insight to what is actually happening on the database server. Firebird/InterBase® statistics should be evaluated regularly and kept, because when things do go wrong, it's immensely helpful to be able to see what they looked like when things were running smoothly. Poor or degrading database performance is practically always to do with poor programming and/or poor transaction handling. The IBExpert Database Statistics retrieves and displays important database statistical information, which can be exported to numerous file formats or printed. This menu item can be found in the IBExpert Services menu.

The Database Statistics are useful for detecting and solving a variety of performance problems, for example, revealing whether an old transaction is still open somewhere, which could be slowing the database down due to the administration of a constantly growing number of record versions. Is the data page fill efficient? Could it be improved by splitting certain large tables into several smaller ones? Or use it to analyze all indices - delete the *bad* and the *useless*, check *actual selectivity* used against the *real selectivity*. And so on and so on.

To generate the current database statistics for a whole database first click on the top left button to view a list of all registered databases:

or simply drag a database node or single or multiple selected table nodes from the Database Explorer to the Database Statistics *Text* page:

to automatically generate statistics for your selection. This method can also be executed using the DB Explorer context-sensitive menu item, *Get statistics for selected tables*.

Alternatively open an existing statistics file to view and analyze statistical records.

If wished, alter the default value Retrieve all Statistics, by selecting one of the following options:

- Stop retrieving after header page statistics
- Stop retrieving after log pages statistics
- Stop retrieving after user indexes statistics
- Stop retrieving after data tables statistics
- Stop retrieving after system tables and indexes statistics

If relevant check the option to analyze the average record and version length (Firebird 1.5, InterBase® 7) which can be found below the toolbar. Then simply click the *Retrieve Statistics* icon (green arrow) to start the retrieval process.

The database's statistical summary is displayed both as text:

as well as in grid form (illustrated in the Tables page section below).

back to top of page

Text page

The text summary provides certain database information (illustration above) as well as a statistical summary broken down by table (illustration below), containing the information also displayed in the grid summary.

The summary displays certain log information, such as the timestamp, page size and ODS version. It then lists the *Oldest transaction*, *Oldest active transaction* (the oldest transaction that has been started but not yet committed or rolled back), Oldest snapshot (this shows where the Garbage Collector will start its work) and Next transaction. These are the statistics you should always keep an eye on as they can indicate a potential source of performance degradation. A large difference between the oldest active transaction (OAT) and the next transaction indicates that there is an open transaction (i.e. a transaction that has been started but not committed) somewhere in the database. Such a problem can cause the database to gradually become slower and slower as the server administrates more and more open versions, and the garbage collection cannot delete older versions. Further information can be found in the Firebird administration using IBExpert chapter, Using the IBExpert Database Statistics.

The Database Statistics display the following information for all tables in the database, both as a log script and in tabular form: *Table name*, *Location*, *Pages, Size* (bytes), *Slots*, *Fill* (%), *DP usage* (%) and *Fill distribution* (an optimal page fill is around 80%). For each table the indices statistics include: Depth, Leaf buckets, Nodes, *Average data length* and *Fill distribution*.

Primary Pointer page: In the illustration above the primary pointer page (PTR) for the EMPLOYEE table is number 172. It begins at the byte that equals the page number 172 multiplied by the page size. This is a sort of table of contents for the EMPLOYEE table; it points to the data pages which contain the table's data.

Index root page: The same information is displayed for the index root pages (IRT) for the indices in this table.

Average record length: This displays how long the data record versions are on average (in bytes). When a dBase table is created, for example, with two fields, each CHAR(100), the average data set length would always be 200. Firebird/InterBase® however does not store adjacent empty spaces. For example a CHAR(100) field containing a string length of 65 followed by 35 empty spaces, is stored by Firebird/InterBase® as a string of 65 plus 1 empty space multiplied by 35. This is why, when data is imported into Firebird/InterBase® from another database, the data is sometimes smaller following the import than it was before. (Please refer to the Database Technology article, The Firebird server and VARCHARs, for further information.)

Total records: How many data sets there are in the individual tables.

Average version length: The length of the record versions on average. When updates are made, you can see here how many bytes on average have altered, compared to the original data set.

Total versions: How many record versions exist for this table. This number should always be as low as possible, as it indicates how many versions of the table Firebird/InterBase® is storing.

Max versions: The maximum number of versions for a record. This indicates that there is one data record that has this number of different versions, which Firebird/InterBase® is having to store because there is still one active transaction somewhere in the database, which prevents old record versions being deleted.

The interesting thing in this case is that this does not only happen for the tables being worked upon, but for all tables. In repeatable read mode, a snapshot is made of the whole database as soon as a transaction is started.

Data pages: How many data pages are used.

Average fill: The amount of data page fill in %.

Fill distribution: The average fill is calculated by how much data is already contained on the data pages. The Firebird/InterBase® server normally fills pages up to a maximum of 80%. The free room is needed for back version storage; if an update to one of the data sets stored on this page is made, the new data set can be stored on the same page as the original version. This saves the number of pages which need to be loaded, should it be necessary to return to the original data set.

The fill distribution also indicates whether the fill for an individual table is an anomaly or if similar problems occur on all tables.

There are certain situations when you might wish for a 100% fill (e.g. when wishing to store an address database on a CD). This can be done with the Use all space option when performing a database restore.

back to top of page Tables page The tables are listed alphabetically by name but, as always in IBExpert, they can be moved or sorted by any of the listed criteria by clicking on the corresponding column header. Column headers can be dragged to the top of the Tables page to display data grouped by that column.

It is possible to calculate certain aggregate functions on the individual columns (see the Fill % column in the illustration above).

The table grid gives some nice feedback about fill and database usage on your tables, e.g. you can quickly spot a table with thousands of pages at 50% fill - wasting half the space and using up cache buffers twice as fast as you could be if the pages were full. This could be an indication of tables with a lot of inserts and deletes, in that case the space will be reused. It could however also be due to a poor page size, e.g. with a page size of 4K or 8K and tables that have perhaps had fields added over a period of time; if the data sets are so large that only one or two records fit onto the page, this will leave a large amount of space.

You may also discover a table, which although covers n data pages using a total of x bytes, with y number of records, but with an average record length of 0. The Versions columns display the same number of records with an average record length of z bytes. This indicates that the table has been deleted and no longer contains any data. However the record versions must still be maintained for old open transactions.

Below the table grid, an index grid displays the statistics for all indices for a selected table. The following information is displayed for indices: Index name, Fields, Unique, Active, Sorting order, Statistics, Depth, Leaf buckets, Nodes, Average data length, Total dup and Fill distribution. Further information can be found under Indices page.

This information in tabular form can be exported (see Export Data) to save the information to file, or printed out.

For further information upon how to use the Database Statistics to maximize database performance, please refer to the Firebird Administration using IBExpert chapter, Using the IBExpert Database Statistics.

back to top of page Indices page In addition to the summary information displayed on the Tables page, the Indices page allows you to analyze all your database indices in depth. Using the drop-down list, you can specify which index types you wish to view:

All indices Bad indices Useless indices Too deep indices Active indices Inactive indices Unique indices Non-unique indices

The indices are listed by table and field but, as always in IBExpert, they can be moved or sorted by any of the listed criteria by clicking on the corresponding column header. Column headers can be dragged to the top of the Indices page to display data grouped by that column. You can immediately discern the index type (Unique, Active, Ascending or Descending).

The Selectivity column displays the actual selectivity which is taken into consideration by the Firebird/InterBase® server, when working out how best to process a query. The Real Selectivity column displays the level of selectivity that could be attained if the index was recomputed. Should you discover discrepancies in these two columns, click the Update selectivity (SET STATISTICS) button to recompute the selectivity. These discrepancies arise because the selectivity is only computed at the time of creation, or when the IBExpert menu item Recompute Selectivity or Recompute All is used (found directly in the Statistic dialog, in the IBExpert Database menu, or in the right-click DB Explorer menu). Alternatively the

SET STATISTIC INDEX {INDEX_NAME} command can be used in the SQL Editor to recompute individual indices.

This is automatically performed during a database backup and restore, as it is not the index, but its definition that is saved, and so the index is therefore reconstructed when the database is restored.

The first thing the Optimizer does when it receives a query is to prepare the execution. It makes decisions regarding indices based solely upon their selectivity. Bad Indices are those considered poor by the Firebird/InterBase® Optimizer. A good selectivity is close to 0 - it's the result of: 1/distinct values. There may be several reasons why the Optimizer may consider a particular index to be bad:

The Optimizer only uses indices with a selectivity < 0.01 if there are no other appropriate indices available. Such an index causes very slow garbage collection in older Firebird and InterBase® versions. This problem has been solved since InterBase® 7.1/7.5 and Firebird 2.0. The index causes the restore process to be very slow, and it is being created extremely slowly (CREATE/ALTER INDEX ACTIVE). This is because the record numbers chain is big for a single index key. If the index is used in a WHERE clause, memory usage will depend on the value being searched (bitmask size). Since the record chain can be large (a lot of key duplicates), memory consumption will be also large. If the index is used in an ORDER BY, and there are a lot of duplicates, mostly in lower key values (depending on the index sort order), there will be lot of index page reads and that will slow down the query. The worst case for an index is when the value in the Uniques column = 1, i.e. all values for an indexed column are the same. These indices are listed as Useless indices. Of course, for your application there may be a situation where such an index is good. For example, if records have an “archive” flag in a column, and your application searches by index on that column only for current, not archived data.

Normally bad and useless indices should be examined and, if not really vital to your application (e.g. if you do not use it to search keys having less duplicates than other keys), deleted. However this is not easy to do if such an index is created by a foreign key because you can only drop it by dropping the foreign key. Dropping the foreign key will however disable the related check constraint, which can be unacceptable. It is possible to replace a foreign key by triggers, but there are some restrictions. Foreign keys control record relations using the index, and the index “sees” all keys for all records independently from the transactions state. A trigger however works only in the client's transaction context. So, when replacing foreign keys with triggers, you must be sure that firstly, records will not be deleted from the master table, or be deleted in a “snapshot table reserving” mode and secondly, ensure that the column used by the primary key in the master table will not ever be modified. You can restrict this using a before update trigger.

If you maintain these conditions, you can drop a particular foreign key.

The next column displays the index depths. An index depth of 2, for example, indicates that Firebird/InterBase® needs to perform two steps to obtain a result. Normally the value should not be higher than three. Should this be the case, a database backup and restore should help.

Leaf buckets display the number of registration leaves, where Firebird/InterBase® can access immediately. Further statistics include nodes, duplicates (total and maximum) and fill distribution.

Further reading: Index Firebird Administration using IBExpert: The Firebird Optimizer and index statistics Firebird Administration using IBExpert: Automating the recalculation of index statistics Recompute selectivity of all indices SQL Editor / Plan Analyzer SQL Editor / Performance Analysis IBExpert Table Editor / Indices Enhancements to indexing in Firebird 2.0 Firebird for the database expert: Episode 1 - Indexes Recreating Indices 1 Recreating Indices 2 back to top of page Blobs page

The Blobs page allows:

The parsing and display of blob statistics (Firebird 3, 4). Additional blob statistics for servers that support the CHAR_LENGTH function. back to top of page Options page It is possible to automatically analyze tables/indices statistics and highlight possible problem tables/indices. This feature is based on the IBEBlock functionality and is therefore is fully customizable.