
BARRODALE COMPUTING SERVICES LTD. (BCS)

DBXten

DBXten (**D**ata**B**ase **e****X**ten**s**ion) is a datatype implementing table-in-table functionality combined with very effective compression strategies. It enables efficient storage and access of very large data sequences and other complex data objects. By utilizing multidimensional indexing and by reducing the number of rows that need to be indexed, DBXten can dramatically speed up ingestion and indexing, while significantly reducing storage space requirements. Advantages of using DBXten for your applications include:

Integration with existing databases. Because DBXten is a modular extension to your existing database server, and it runs on your existing hardware, your investment in licenses, software, hardware and training is preserved.

Universal data type. DBXten's table-in-table concept allows you to store a diverse range of data without having to provide code for custom types. For example, DBXten can store time series, N-dimensional grids, molecular structures, raster images, and geometrical primitives (e.g., polygons, 3D points, meshes, etc.) with any number of attributes.

Automated compression. DBXten's compression strategies work behind the scenes to do more than dramatically reduce storage costs and network traffic – they transform the table-in-table notion from a concept to an invaluable tool. DBXten automatically analyzes each column of your data and picks a compression strategy that's right for it. Depending on the information content, DBXten may deflate a column to a small fraction of its original size, or even to just a few bytes (regardless of the number of rows).

Flexible resource management. DBXten's client libraries allow client machines to take most of the load off your database server, both when inserting and extracting data. In an era where the price of commodity workstations keeps dropping but database license fees remain flat, DBXten allows you to make the most of your server investment.

Powerful indexing. DBXten combines the space efficiency of aggregate indexes with the flexibility of functional indexes, allowing you to index both stored and derived values. For example, given stored multiband satellite data, a functional index on sea surface temperature (SST – a derived quantity) could be generated, without storing the computed SST values.

Experienced programmers know that the greatest gains in efficiency typically come from replacing the algorithm, rather than from tweaking existing code. DBXten is the most effective algorithm available for efficiently handling very large collections of complex data in database applications. Please contact BCSinfo@barrodale.com for further information about this patent pending BCS database extension.