Query Processor API

QueryProcessor is used to parse, optimize, and then execute queries. The parse step turns a query into relational algebra, the optimize step selects the optimal ordering of operations, with the intent to minimize disc accesses, and the execution step performs each individual operation, resulting in a relation based on the initial query.

QueryProcessor()

Initializes access to the database, and begins listening for evaluation requests

~QueryProcessor()

Halt listening for evaluation requests.

Smarterator™ evaluateQuery(string query, int sessionId)

Corresponding to the parse step of query processing, this function takes in a query to be processed and parses it, sending corresponding relational algebra to the Optimizer in the form of an expression tree.

This function makes no assumptions about the validity of the query string, and throws an exception if it fails to parse the string. The sessionId is used to find relevant session information for security and priority. Higher priorities are used for determining priority within the BufferManager while the security information determines read/write access to portions of the database, as well as other privileges.

The resulting Smarterator™ describes the final relation generated by the query.