**DB Access Class – (dbAccess)**

**Attributes**

* Linked – private – MySQL connection flag, holds Link Identifier if Alive
* Result – private – Holds query results as an Associative Array or
* InjectID – private – Holds the last insert auto incremented ID
* DB\_SQL – private – Holds most recent SQL statement.

**Methods**

* \_\_construct() – Initializes default attribute values
* Result() – Returns the results from the last query run as an associative array or as an array of associative arrays.
  + ACCEPTS:
    - void
  + RETURNS:
    - Associative Array | Array( Assoc. Array )
* LastQuery() – Returns the most recently run SQL statement
  + ACCEPTS:
    - void
  + RETURNS:
    - String, SQL statement
* InjectID() – Returns the most recent insert auto incremented ID
  + ACCEPTS:
    - void
  + RETURNS:
    - ID
* LinkID() – Returns the current Link Identifier
  + ACCEPTS:
    - void
  + RETURNS:
    - Link Identifier
* Link() – Establishes a connection to the database
  + ACCEPTS:
    - void
  + RETURNS:
    - True | False
* SelectDB() – Selects a database to use. If the database does not exist this method will attempt to create the database.
  + ACCEPTS:
    - $db\_name – The name of the database to select
  + RETURNS:
    - True | False
* TableExists() – Checks whether or not a table exists within the selected database
  + ACCEPTS:
    - $tableName – The name of the table to check for.
  + RETURNS:
    - True | False
* Snatch() – Performs a MySQL select statement based on the provided details and populates the Result attribute with the results of the select query. If the query was successful but no results were returned the Error attribute will be set to “0 results found” and this method will return false, since no data was actually snatched.
  + ACCEPTS:
    - $tableName – The name of the table to snatch from
    - $field – The field/column name or array of names to snatch, default = “\*”
    - $where – The where clause or array of where clauses to apply
    - $order – The order by clause to apply
    - $distinct – Boolean, distinct values flag
    - $limit – The limit of results to return
  + RETURNS:
    - True | False
* Gather() – Performs a MySQL join statement based on the provided details and populates the Result attribute with the results of the join query. If the query was successful but no results were returned the Error attribute will be set to “0 results found” and this method will return false, since no data was actually gathered. This method only accepts multiple tables, if one table use Snatch() method.
  + ACCEPTS:
    - $tables – The names of the tables to snatch from, must be array!
    - $linkColumn – The field/column name that is common among all tables
    - $where – The where clause or array of where clauses to apply
    - $fields – The field/column name or array of names to gather, default = “\*”
    - $method – The method to use for joining the table data, default = “LEFT”. {INNER, RIGHT, FULL}
  + RETURNS:
    - True | False
* Scatter() – Performs a multiple MySQL insert or update queries, NOT WRITTEN
* Inject() – Performs a MySQL insert statement based on the provided details and populates the Result attribute with the number of affected rows. The insert ID is then stored into the InjectID attribute.
  + ACCEPTS:
    - $table – The name of the table to insert into
    - $column – The field/column name or array of names to insert values for
    - $value – The value or array of values to insert
    - $flag – The insert mode flag to switch between the different sorts of inserts, default = “normal”. {EMPTY, DELAYED, UPDATE}
  + RETURNS:
    - True | False
* Refresh() – Performs a MySQL update statement based on the provided details and populates the Result attribute with the number of affected rows. If the query was successful but no rows were affected the Error attribute will be set to “0 rows affected” and this method will return false, since no data was actually updated.
  + ACCEPTS:
    - $table – The name of the table to update
    - $set – The field/column name or array of names to update values for
    - $value – The value or array of values to update with
    - $where – The where clause or array of where clauses to apply
  + RETURNS:
    - True | False
* NewTable() – Creates a new empty table using the provided table name and field/column descriptions.
  + ACCEPTS:
    - $tableName – The name of the table to create
    - $fieldList – The description of the field or array of descriptions to structure the table with.
  + RETURNS:
    - True | False
* RenameTable() – Renames the table or tables listed using the new name or names provided.
  + ACCEPTS:
    - $tableName – The name or array of names of tables to rename
    - $newName – The new name or array of new names to give the table/s
  + RETURNS:
    - True | False
* ModTable() – Modifies database table fields/columns in several different ways.
  + ACCEPTS:
    - $tableName – The name of the table to modify
    - $what – The method of modification.
      * ADD – default, adds a new field to the table
      * TYPE – Changes a fields datatype
      * RENAME – Renames a field
      * DROP – Drops a field.
    - $colName – name or names of fields to modify. May only be array for “DROP” or “ADD” methods.
    - $type – the MySQL datatype to use for the given field. May only be array when using the “ADD” method.
    - $value – the default value of the field/column OR the newName if using the “RENAME” method. May only be array when using “ADD” method.
  + RETURNS:
    - True | False
* Kill() – Deletes tables/table rows & databases, if only a name is given the method will return false in an effort to avoid accidental data wipe outs. Must set $where = null if setting a $drop flag.
  + ACCEPTS:
    - $T\_DB – The name of the table or database to kill
    - $where – The where clause or array of where clauses to apply (rows only)
    - $drop – The method of killing to use.
      * “database” – flag to drop a database
      * “truncate” – flag to truncate a given table (resets increment, clears)
      * “clear” – flag to clear a table of all rows (no increment reset)
      * “drop” – flag to drop a table
  + RETURNS:
    - True | False
* RawQuery() – This method is here for last resort. You must compile the SQL statements properly and clean the data before sending it. It is recommended to use the other methods in the class if possible. The Error attribute is populated with error information, if any exists.
  + ACCEPTS:
    - $SQL – The user provided SQL statement to run
  + RETURNS:
    - Array(Boolean, #Affected Rows)
* Sever() – Severs the connection to the database, if one exists
  + ACCEPTS:
    - void
  + RETURNS:
    - True | False
* Purify() – Converts raw input data into html special characters, strips tags, and real escapes strings, then returns the clean value.
  + ACCEPTS:
    - $value – the value to be purified
  + RETURNS:
    - Purified value
* Decrypt() – Decrypts only database values that have been encrypted by \*this unique installation of the NiFrame system and returns the value.
  + ACCEPTS:
    - $string – string to be decrypted
  + RETURNS:
    - Decrypted string