Оглавление

[**DK\_SQLite3** 2](#_Toc112480360)

[TSQLite3 2](#_Toc112480361)

# **DK\_SQLite3**

ЗАВИСИТ ОТ МОДУЛЕЙ: **DKUtils.DK\_Vector, DKUtils.DK\_SQLUtils, DKUtils.DK\_StrUtils**

## TSQLite3

Работа с SQLite 3

Методы:

|  |  |
| --- | --- |
| procedure Connect(const AFileName: String); | Создает соединение с базой данных в файле с именем AFileName |
| procedure ExecuteScript(const AFileName: String); | Запускает SQL скрипт, загруженный из файла AFileName |
| function LastWritedInt32ID(const ATableName: String): Integer;  function LastWritedInt64ID(const ATableName: String): Int64; | Возвращает последнее записанное в таблицу ATableName значение первичного ключа |
| function LastWritedInt32Value(const ATableName, AFieldName: String): Integer;  function LastWritedInt64Value(const ATableName, AFieldName: String): Int64;  function LastWritedStringValue(const ATableName, AFieldName: String): String;  function LastWritedDateTimeValue(const ATableName, AFieldName: String): TDateTime; | Возвращает последнее записанное в таблицу ATableName в поле AFieldName значение |
| function IsValueInTable(const ATableName, AFieldName: String;  const AValue: Integer): Boolean;  function IsValueInTable(const ATableName, AFieldName: String;  const AValue: Int64): Boolean;  function IsValueInTable(const ATableName, AFieldName: String;  const AValue: TDateTime): Boolean;  function IsValueInTable(const ATableName, AFieldName: String;  const AValue: String; const ACaseSensitivity: Boolean = True): Boolean; | Проверка наличия в таблице ATableName в поле AFieldName значения AValue.  ACaseSensitivity – чувствительность к регистру строкового значения |
| procedure KeyPickList(const ATableName, AKeyFieldName, APickFieldName: String;  out AKeyVector: TIntVector;  out APickVector: TStrVector;  const AKeyValueNotZero: Boolean = False;  const AOrderFieldName: String = '');  procedure KeyPickList(const ATableName, AKeyFieldName, APickFieldName: String;  out AKeyList, APickList: TStringList;  const AKeyValueNotZero: Boolean = False;  const AOrderFieldName: String = ''); | Получает из таблицы ATableName значения из полей AKeyFieldName и APickFieldName, отсортированные по AOrderFieldName (если AOrderFieldName=EmptyStr, то AOrderFieldName=APickFieldName), и записывает их в AKeyVector,APickVector или AKeyList,APickList.  Если AKeyValueNotZero=True - отбор значений AKeyFieldName<>0 |
| function ValueInt32Int32ID(const ATableName, AValueFieldName, AIDFieldName: String;  const AIDValue: Integer): Integer;  function ValueInt64Int32ID(const ATableName, AValueFieldName, AIDFieldName: String;  const AIDValue: Integer): Int64;  function ValueDTInt32ID(const ATableName, AValueFieldName, AIDFieldName: String;  const AIDValue: Integer): TDateTime;  function ValueStrInt32ID(const ATableName, AValueFieldName, AIDFieldName: String;  const AIDValue: Integer): String;  function ValueInt32Int64ID(const ATableName, AValueFieldName, AIDFieldName: String;  const AIDValue: Int64): Integer;  function ValueInt64Int64ID(const ATableName, AValueFieldName, AIDFieldName: String;  const AIDValue: Int64): Int64;  function ValueDTInt64ID(const ATableName, AValueFieldName, AIDFieldName: String;  const AIDValue: Int64): TDateTime;  function ValueStrInt64ID(const ATableName, AValueFieldName, AIDFieldName: String;  const AIDValue: Int64): String; | Получает из таблицы ATableName значение поля AValueFieldName, для которого значение ключевого поля AIDFieldName равно AIDValue. Возвращает полученное значение. |
| procedure Delete(const ATableName, AIDFieldName: String; const AIDValue: String);  procedure Delete(const ATableName, AIDFieldName: String; const AIDValue: Integer);  procedure Delete(const ATableName, AIDFieldName: String; const AIDValue: Int64);  procedure Delete(const ATableName, AIDFieldName: String; const AIDValue: TDateTime); | Удаляет из таблицы ATable значения, для которых AIDField=AIDValue |
| procedure UpdateInt32ID(const ATableName, AFieldName, AIDFieldName: String;  const AIDValue: Integer;  const ANewValue: Integer);  procedure UpdateInt32ID(const ATableName, AFieldName, AIDFieldName: String;  const AIDValue: Integer;  const ANewValue: Int64);  procedure UpdateInt32ID(const ATableName, AFieldName, AIDFieldName: String;  const AIDValue: Integer;  const ANewValue: TDateTime);  procedure UpdateInt32ID(const ATableName, AFieldName, AIDFieldName: String;  const AIDValue: Integer;  const ANewValue: String);  procedure UpdateInt64ID(const ATableName, AFieldName, AIDFieldName: String;  const AIDValue: Int64;  const ANewValue: Integer);  procedure UpdateInt64ID(const ATableName, AFieldName, AIDFieldName: String;  const AIDValue: Int64;  const ANewValue: Int64);  procedure UpdateInt64ID(const ATableName, AFieldName, AIDFieldName: String;  const AIDValue: Int64;  const ANewValue: TDateTime);  procedure UpdateInt64ID(const ATableName, AFieldName, AIDFieldName: String;  const AIDValue: Int64;  const ANewValue: String); | Записывает в поле AFieldName таблицы ATableName значение ANewValue в строку, для которой значение идентификатора в поле AIDFieldName равно AIDValue |
| function IsValueInTableNotMatchInt32ID(const ATableName, AFieldName: String;  const AValue: Integer;  const AIDFieldName: String;  const AIDValue: Integer = 0): Boolean;  function IsValueInTableNotMatchInt32ID(const ATableName, AFieldName: String;  const AValue: Int64;  const AIDFieldName: String;  const AIDValue: Integer = 0): Boolean;  function IsValueInTableNotMatchInt32ID(const ATableName, AFieldName: String;  const AValue: TDateTime;  const AIDFieldName: String;  const AIDValue: Integer = 0): Boolean;  function IsValueInTableNotMatchInt32ID(const ATableName, AFieldName: String;  const AValue: String;  const AIDFieldName: String;  const AIDValue: Integer = 0;  const ACaseSensitivity: Boolean = True): Boolean;  function IsValueInTableNotMatchInt64ID(const ATableName, AFieldName: String;  const AValue: Integer;  const AIDFieldName: String;  const AIDValue: Int64 = 0): Boolean;  function IsValueInTableNotMatchInt64ID(const ATableName, AFieldName: String;  const AValue: Int64;  const AIDFieldName: String;  const AIDValue: Int64 = 0): Boolean;  function IsValueInTableNotMatchInt64ID(const ATableName, AFieldName: String;  const AValue: TDateTime;  const AIDFieldName: String;  const AIDValue: Int64 = 0): Boolean;  function IsValueInTableNotMatchInt64ID(const ATableName, AFieldName: String;  const AValue: String;  const AIDFieldName: String;  const AIDValue: Int64 = 0;  const ACaseSensitivity: Boolean = True): Boolean; | Проверяет наличие в поле AFieldName таблицы ATableName значения AValue, у которого идентификатор в поле AIDFieldName не равен значению AIDValue  (Проверка на дублирующееся значение) |