MOBILE_ERP_STORE DATABASE API

1.0

Generated by Doxygen 1.9.1

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

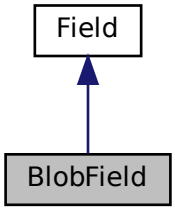
Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

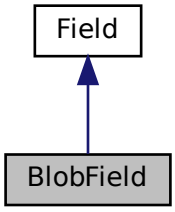
Class Documentation

3.1 BlobField Class Reference

Inheritance diagram for BlobField:



Collaboration diagram for BlobField:



Public Member Functions

- BlobField ()

  default constructor.
- BlobField (string name, FieldType type=Blob)

  Parameterized constructor.
- ~BlobField ()

  destructor.

Additional Inherited Members

3.1.1 Constructor & Destructor Documentation

3.1.1.1 BlobField()

```
BlobField::BlobField (
            string name,
            FieldType type = Blob )
```

Parameterized constructor.

Parameters

name	name of one of the database
FieldType	this is the type of the database (default value is Blob)

The documentation for this class was generated from the following file:

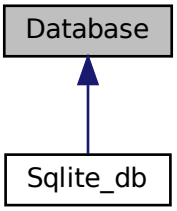
- include/sqlite_database.hpp

3.2 Database Class Reference

The Database class is an abstract class supposed to encapsulate management of SQL based database systems.

```
#include <database.hpp>
```

Inheritance diagram for Database:



Public Member Functions

- virtual string **get_uri** ()=0
- virtual string **get_name** ()=0
- virtual DatabaseType **get_type** ()=0
- virtual int32_t **create** ()=0
- virtual int32_t **drop** ()=0

Protected Attributes

- string **name**
- string **uri**
- DatabaseType **type**

3.2.1 Detailed Description

The Database class is an abstract class supposed to encapsulate management of SQL based database systems.

This class represents database systems using SQL as definition and manipulation language. The class can be used as model for Object-Relational-Mapping in an MVC architecture The currently supported database systems are

- Sqlite
- MySql,
- PostGreSql

Database abstract class provides the following services

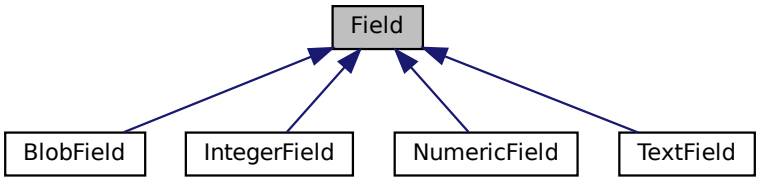
- create – creates the database in the DB management systems
- drop – drops the database in the DB management systems

The documentation for this class was generated from the following file:

- include/database.hpp

3.3 Field Class Reference

Inheritance diagram for Field:



Public Member Functions

- virtual string **get_name** ()=0
- virtual char **get_char** ()=0
- virtual int **get_int** ()=0
- virtual float **get_float** ()=0
- virtual string **get_string** ()=0

Protected Member Functions

- virtual FieldType **get_type** ()=0
- virtual string **get_create_stmt_sql** ()=0

Protected Attributes

- string **name**
- istream **value**
- FieldType **type**
- vector< ConstraintType > **constraints**

The documentation for this class was generated from the following file:

- include/database.hpp

3.4 fts5_api Struct Reference

Public Attributes

- int **iVersion**
- int(∗ **xCreateTokenizer**)(fts5_api ∗pApi, const char ∗zName, void ∗pContext, fts5_tokenizer ∗pTokenizer, void(∗xDestroy)(void ∗))
- int(∗ **xFindTokenizer**)(fts5_api ∗pApi, const char ∗zName, void ∗∗ppContext, fts5_tokenizer ∗pTokenizer)
- int(∗ **xCreateFunction**)(fts5_api ∗pApi, const char ∗zName, void ∗pContext, fts5_extension_function x↩ Function, void(∗xDestroy)(void ∗))

The documentation for this struct was generated from the following file:

- include/sqlite3.h

3.5 fts5_tokenizer Struct Reference

Public Attributes

- int(∗ **xCreate**)(void ∗, const char ∗∗azArg, int nArg, Fts5Tokenizer ∗∗ppOut)
- void(∗ **xDelete**)(Fts5Tokenizer ∗)
- int(∗ **xTokenize**)(Fts5Tokenizer ∗, void ∗pCtx, int flags, const char ∗pText, int nText, int(∗xToken)(void ∗pCtx, int tflags, const char ∗pToken, int nToken, int iStart, int iEnd))

The documentation for this struct was generated from the following file:

- include/sqlite3.h

3.6 Fts5ExtensionApi Struct Reference

Public Attributes

- int **iVersion**
- void ∗(∗ **xUserData**)(Fts5Context ∗)
- int(∗ **xColumnCount**)(Fts5Context ∗)
- int(∗ **xRowCount**)(Fts5Context ∗, sqlite3_int64 ∗pnRow)
- int(∗ **xColumnTotalSize**)(Fts5Context ∗, int iCol, sqlite3_int64 ∗pnToken)
- int(∗ **xTokenize**)(Fts5Context ∗, const char ∗pText, int nText, void ∗pCtx, int(∗xToken)(void ∗, int, const char ∗, int, int, int))
- int(∗ **xPhraseCount**)(Fts5Context ∗)
- int(∗ **xPhraseSize**)(Fts5Context ∗, int iPhrase)
- int(∗ **xInstCount**)(Fts5Context ∗, int ∗pnInst)
- int(∗ **xInst**)(Fts5Context ∗, int iIdx, int ∗piPhrase, int ∗piCol, int ∗piOff)
- sqlite3_int64(∗ **xRowid**)(Fts5Context ∗)
- int(∗ **xColumnText**)(Fts5Context ∗, int iCol, const char ∗∗pz, int ∗pn)
- int(∗ **xColumnSize**)(Fts5Context ∗, int iCol, int ∗pnToken)
- int(∗ **xQueryPhrase**)(Fts5Context ∗, int iPhrase, void ∗pUserData, int(∗)(const Fts5ExtensionApi ∗, Fts5↩ Context ∗, void ∗))
- int(∗ **xSetAuxdata**)(Fts5Context ∗, void ∗pAux, void(∗xDelete)(void ∗))
- void ∗(∗ **xGetAuxdata**)(Fts5Context ∗, int bClear)
- int(∗ **xPhraseFirst**)(Fts5Context ∗, int iPhrase, Fts5PhraseIter ∗, int ∗, int ∗)
- void(∗ **xPhraseNext**)(Fts5Context ∗, Fts5PhraseIter ∗, int ∗piCol, int ∗piOff)
- int(∗ **xPhraseFirstColumn**)(Fts5Context ∗, int iPhrase, Fts5PhraseIter ∗, int ∗)
- void(∗ **xPhraseNextColumn**)(Fts5Context ∗, Fts5PhraseIter ∗, int ∗piCol)

The documentation for this struct was generated from the following file:

- include/sqlite3.h

3.7 Fts5PhraseIter Struct Reference

Public Attributes

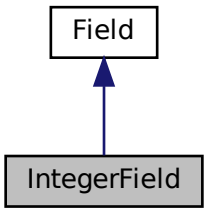
- const unsigned char * **a**
- const unsigned char * **b**

The documentation for this struct was generated from the following file:

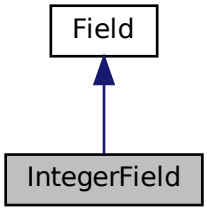
- include/sqlite3.h

3.8 IntegerField Class Reference

Inheritance diagram for IntegerField:



Collaboration diagram for IntegerField:



Public Member Functions

- IntegerField ()
  default constructor.
- IntegerField (string name, FieldType type=Integer)
  Parameterized constructor.
- ~IntegerField ()
  destructor.

Additional Inherited Members

3.8.1 Constructor & Destructor Documentation

3.8.1.1 IntegerField()

```
IntegerField::IntegerField (
            string name,
            FieldType type = Integer )
```

Parameterized constructor.

Parameters

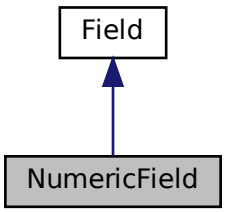
name	name of one of the database
FieldType	this is the type of the database (default value is Integer)

The documentation for this class was generated from the following file:

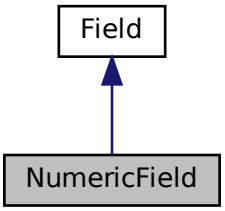
- include/sqlite_database.hpp

3.9 NumericField Class Reference

Inheritance diagram for NumericField:



Collaboration diagram for NumericField:



Public Member Functions

- NumericField ()
  default constructor.
- NumericField (string name, FieldType type=Numeric)
  Parameterized constructor.
- ~NumericField ()
  destructor.

Additional Inherited Members

3.9.1 Constructor & Destructor Documentation

3.9.1.1 NumericField()

```
NumericField::NumericField (
            string name,
            FieldType type = Numeric )
```

Parameterized constructor.

Parameters

name	name of one of the database
FieldType	this is the type of the database (default value is Numeric)

The documentation for this class was generated from the following file:

- include/sqlite_database.hpp

3.10 Query Class Reference

Public Member Functions

- virtual string **get_db_name** ()=0
- virtual string **get_tbl_name** ()=0
- virtual QueryType **get_type** ()=0
- virtual int32_t **execute** ()=0

Protected Member Functions

- virtual string **get_sql_stmt** ()=0

Protected Attributes

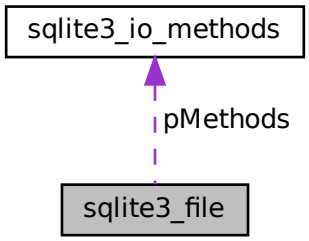
- string **db_name**
- string **tbl_name**
- string **sql_stmt**
- QueryType **type**

The documentation for this class was generated from the following file:

- include/database.hpp

3.11 sqlite3_file Struct Reference

Collaboration diagram for sqlite3_file:



Public Attributes

- const struct sqlite3_io_methods * **pMethods**

The documentation for this struct was generated from the following file:

- include/sqlite3.h

3.12 sqlite3_index_info::sqlite3_index_constraint Struct Reference

Public Attributes

- int **iColumn**
- unsigned char **op**
- unsigned char **usable**
- int **iTermOffset**

The documentation for this struct was generated from the following file:

- include/sqlite3.h

3.13 sqlite3_index_info::sqlite3_index_constraint_usage Struct Reference

Public Attributes

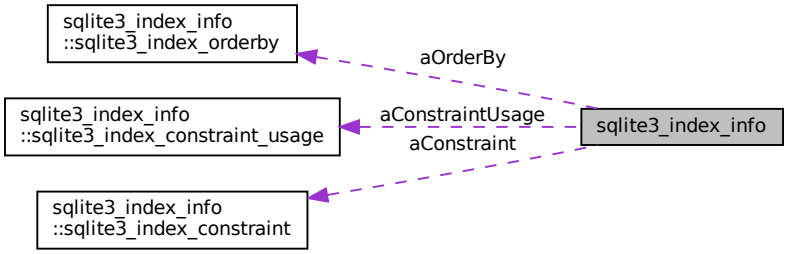
- int **argvIndex**
- unsigned char **omit**

The documentation for this struct was generated from the following file:

- include/sqlite3.h

3.14 sqlite3_index_info Struct Reference

Collaboration diagram for sqlite3_index_info:



Classes

- struct sqlite3_index_constraint
- struct sqlite3_index_constraint_usage
- struct sqlite3_index_orderby

Public Attributes

- int **nConstraint**
- struct sqlite3_index_info::sqlite3_index_constraint * **aConstraint**
- int **nOrderBy**
- struct sqlite3_index_info::sqlite3_index_orderby * **aOrderBy**
- struct sqlite3_index_info::sqlite3_index_constraint_usage * **aConstraintUsage**
- int **idxNum**
- char * **idxStr**
- int **needToFreeIdxStr**
- int **orderByConsumed**
- double **estimatedCost**
- sqlite3_int64 **estimatedRows**
- int **idxFlags**
- sqlite3_uint64 **colUsed**

The documentation for this struct was generated from the following file:

- include/sqlite3.h

3.15 sqlite3_index_info::sqlite3_index_orderby Struct Reference

Public Attributes

- int **iColumn**
- unsigned char **desc**

The documentation for this struct was generated from the following file:

- include/sqlite3.h

3.16 sqlite3_io_methods Struct Reference

Public Attributes

- int **iVersion**
- int(* **xClose**)(sqlite3_file *)
- int(* **xRead**)(sqlite3_file *, void *, int iAmt, sqlite3_int64 iOfst)
- int(* **xWrite**)(sqlite3_file *, const void *, int iAmt, sqlite3_int64 iOfst)
- int(* **xTruncate**)(sqlite3_file *, sqlite3_int64 size)
- int(* **xSync**)(sqlite3_file *, int flags)
- int(* **xFileSize**)(sqlite3_file *, sqlite3_int64 *pSize)
- int(* **xLock**)(sqlite3_file *, int)
- int(* **xUnlock**)(sqlite3_file *, int)
- int(* **xCheckReservedLock**)(sqlite3_file *, int *pResOut)
- int(* **xFileControl**)(sqlite3_file *, int op, void *pArg)
- int(* **xSectorSize**)(sqlite3_file *)
- int(* **xDeviceCharacteristics**)(sqlite3_file *)
- int(* **xShmMap**)(sqlite3_file *, int iPg, int pgsz, int, void volatile **)
- int(* **xShmLock**)(sqlite3_file *, int offset, int n, int flags)
- void(* **xShmBarrier**)(sqlite3_file *)
- int(* **xShmUnmap**)(sqlite3_file *, int deleteFlag)
- int(* **xFetch**)(sqlite3_file *, sqlite3_int64 iOfst, int iAmt, void **pp)
- int(* **xUnfetch**)(sqlite3_file *, sqlite3_int64 iOfst, void *p)

The documentation for this struct was generated from the following file:

- include/sqlite3.h

3.17 sqlite3_mem_methods Struct Reference

Public Attributes

- void ∗(∗ **xMalloc**)(int)
- void(∗ **xFree**)(void ∗)
- void ∗(∗ **xRealloc**)(void ∗, int)
- int(∗ **xSize**)(void ∗)
- int(∗ **xRoundup**)(int)
- int(∗ **xInit**)(void ∗)
- void(∗ **xShutdown**)(void ∗)
- void ∗ **pAppData**

The documentation for this struct was generated from the following file:

- include/sqlite3.h

3.18 sqlite3_module Struct Reference

Public Attributes

- int **iVersion**
- int(∗ **xCreate**)(sqlite3 ∗, void ∗pAux, int argc, const char ∗const ∗argv, sqlite3_vtab ∗∗ppVTab, char ∗∗)
- int(∗ **xConnect**)(sqlite3 ∗, void ∗pAux, int argc, const char ∗const ∗argv, sqlite3_vtab ∗∗ppVTab, char ∗∗)
- int(∗ **xBestIndex**)(sqlite3_vtab ∗pVTab, sqlite3_index_info ∗)
- int(∗ **xDisconnect**)(sqlite3_vtab ∗pVTab)
- int(∗ **xDestroy**)(sqlite3_vtab ∗pVTab)
- int(∗ **xOpen**)(sqlite3_vtab ∗pVTab, sqlite3_vtab_cursor ∗∗ppCursor)
- int(∗ **xClose**)(sqlite3_vtab_cursor ∗)
- int(∗ **xFilter**)(sqlite3_vtab_cursor ∗, int idxNum, const char ∗idxStr, int argc, sqlite3_value ∗∗argv)
- int(∗ **xNext**)(sqlite3_vtab_cursor ∗)
- int(∗ **xEof**)(sqlite3_vtab_cursor ∗)
- int(∗ **xColumn**)(sqlite3_vtab_cursor ∗, sqlite3_context ∗, int)
- int(∗ **xRowid**)(sqlite3_vtab_cursor ∗, sqlite3_int64 ∗pRowid)
- int(∗ **xUpdate**)(sqlite3_vtab ∗, int, sqlite3_value ∗∗, sqlite3_int64 ∗)
- int(∗ **xBegin**)(sqlite3_vtab ∗pVTab)
- int(∗ **xSync**)(sqlite3_vtab ∗pVTab)
- int(∗ **xCommit**)(sqlite3_vtab ∗pVTab)
- int(∗ **xRollback**)(sqlite3_vtab ∗pVTab)
- int(∗ **xFindFunction**)(sqlite3_vtab ∗pVtab, int nArg, const char ∗zName, void(∗∗pxFunc)(sqlite3_context ∗, int, sqlite3_value ∗∗), void ∗∗ppArg)
- int(∗ **xRename**)(sqlite3_vtab ∗pVtab, const char ∗zNew)
- int(∗ **xSavepoint**)(sqlite3_vtab ∗pVTab, int)
- int(∗ **xRelease**)(sqlite3_vtab ∗pVTab, int)
- int(∗ **xRollbackTo**)(sqlite3_vtab ∗pVTab, int)
- int(∗ **xShadowName**)(const char ∗)

The documentation for this struct was generated from the following file:

- include/sqlite3.h

3.19 sqlite3_mutex_methods Struct Reference

Public Attributes

- int(∗ **xMutexInit**)(void)
- int(∗ **xMutexEnd**)(void)
- sqlite3_mutex ∗(∗ **xMutexAlloc**)(int)
- void(∗ **xMutexFree**)(sqlite3_mutex ∗)
- void(∗ **xMutexEnter**)(sqlite3_mutex ∗)
- int(∗ **xMutexTry**)(sqlite3_mutex ∗)
- void(∗ **xMutexLeave**)(sqlite3_mutex ∗)
- int(∗ **xMutexHeld**)(sqlite3_mutex ∗)
- int(∗ **xMutexNotheld**)(sqlite3_mutex ∗)

The documentation for this struct was generated from the following file:

- include/sqlite3.h

3.20 sqlite3_pcache_methods Struct Reference

Public Attributes

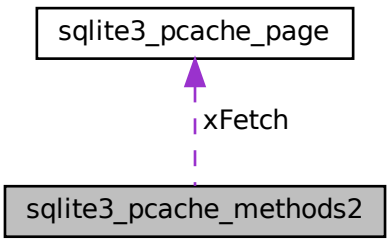
- void ∗ **pArg**
- int(∗ **xInit**)(void ∗)
- void(∗ **xShutdown**)(void ∗)
- sqlite3_pcache ∗(∗ **xCreate**)(int szPage, int bPurgeable)
- void(∗ **xCachesize**)(sqlite3_pcache ∗, int nCachesize)
- int(∗ **xPagecount**)(sqlite3_pcache ∗)
- void ∗(∗ **xFetch**)(sqlite3_pcache ∗, unsigned key, int createFlag)
- void(∗ **xUnpin**)(sqlite3_pcache ∗, void ∗, int discard)
- void(∗ **xRekey**)(sqlite3_pcache ∗, void ∗, unsigned oldKey, unsigned newKey)
- void(∗ **xTruncate**)(sqlite3_pcache ∗, unsigned iLimit)
- void(∗ **xDestroy**)(sqlite3_pcache ∗)

The documentation for this struct was generated from the following file:

- include/sqlite3.h

3.21 sqlite3_pcache_methods2 Struct Reference

Collaboration diagram for sqlite3_pcache_methods2:



Public Attributes

- int **iVersion**
- void ∗ **pArg**
- int(∗ **xInit**)(void ∗)
- void(∗ **xShutdown**)(void ∗)
- sqlite3_pcache ∗(∗ **xCreate**)(int szPage, int szExtra, int bPurgeable)
- void(∗ **xCachesize**)(sqlite3_pcache ∗, int nCachesize)
- int(∗ **xPagecount**)(sqlite3_pcache ∗)
- sqlite3_pcache_page ∗(∗ **xFetch**)(sqlite3_pcache ∗, unsigned key, int createFlag)
- void(∗ **xUnpin**)(sqlite3_pcache ∗, sqlite3_pcache_page ∗, int discard)
- void(∗ **xRekey**)(sqlite3_pcache ∗, sqlite3_pcache_page ∗, unsigned oldKey, unsigned newKey)
- void(∗ **xTruncate**)(sqlite3_pcache ∗, unsigned iLimit)
- void(∗ **xDestroy**)(sqlite3_pcache ∗)
- void(∗ **xShrink**)(sqlite3_pcache ∗)

The documentation for this struct was generated from the following file:

- include/sqlite3.h

3.22 sqlite3_pcache_page Struct Reference

Public Attributes

- void ∗ **pBuf**
- void ∗ **pExtra**

The documentation for this struct was generated from the following file:

- include/sqlite3.h

3.23 sqlite3_rtree_geometry Struct Reference

Public Attributes

- void ∗ **pContext**
- int **nParam**
- sqlite3_rtree_dbl ∗ **aParam**
- void ∗ **pUser**
- void(∗ **xDelUser**)(void ∗)

The documentation for this struct was generated from the following file:

- include/sqlite3.h

3.24 sqlite3_rtree_query_info Struct Reference

Public Attributes

- void * **pContext**
- int **nParam**
- sqlite3_rtree_dbl * **aParam**
- void * **pUser**
- void(* **xDelUser**)(void *)
- sqlite3_rtree_dbl * **aCoord**
- unsigned int * **anQueue**
- int **nCoord**
- int **iLevel**
- int **mxLevel**
- sqlite3_int64 **iRowid**
- sqlite3_rtree_dbl **rParentScore**
- int **eParentWithin**
- int **eWithin**
- sqlite3_rtree_dbl **rScore**
- sqlite3_value ** **apSqlParam**

The documentation for this struct was generated from the following file:

- include/sqlite3.h

3.25 sqlite3_snapshot Struct Reference

Public Attributes

- unsigned char **hidden** [48]

The documentation for this struct was generated from the following file:

- include/sqlite3.h

3.26 sqlite3_vfs Struct Reference

Collaboration diagram for sqlite3_vfs:

Public Attributes

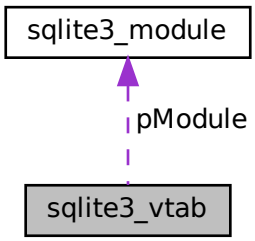
- int **iVersion**
- int **szOsFile**
- int **mxPathname**
- sqlite3_vfs * **pNext**
- const char * **zName**
- void * **pAppData**
- int(* **xOpen**)(sqlite3_vfs *, sqlite3_filename zName, sqlite3_file *, int flags, int *pOutFlags)
- int(* **xDelete**)(sqlite3_vfs *, const char *zName, int syncDir)
- int(* **xAccess**)(sqlite3_vfs *, const char *zName, int flags, int *pResOut)
- int(* **xFullPathname**)(sqlite3_vfs *, const char *zName, int nOut, char *zOut)
- void *(* **xDlOpen**)(sqlite3_vfs *, const char *zFilename)
- void(* **xDlError**)(sqlite3_vfs *, int nByte, char *zErrMsg)
- void(*(* **xDlSym**)(sqlite3_vfs *, void *, const char *zSymbol))(void)
- void(* **xDlClose**)(sqlite3_vfs *, void *)
- int(* **xRandomness**)(sqlite3_vfs *, int nByte, char *zOut)
- int(* **xSleep**)(sqlite3_vfs *, int microseconds)
- int(* **xCurrentTime**)(sqlite3_vfs *, double *)
- int(* **xGetLastError**)(sqlite3_vfs *, int, char *)
- int(* **xCurrentTimeInt64**)(sqlite3_vfs *, sqlite3_int64 *)
- int(* **xSetSystemCall**)(sqlite3_vfs *, const char *zName, sqlite3_syscall_ptr)
- sqlite3_syscall_ptr(* **xGetSystemCall**)(sqlite3_vfs *, const char *zName)
- const char *(* **xNextSystemCall**)(sqlite3_vfs *, const char *zName)

The documentation for this struct was generated from the following file:

- include/sqlite3.h

3.27 sqlite3_vtab Struct Reference

Collaboration diagram for sqlite3_vtab:



Public Attributes

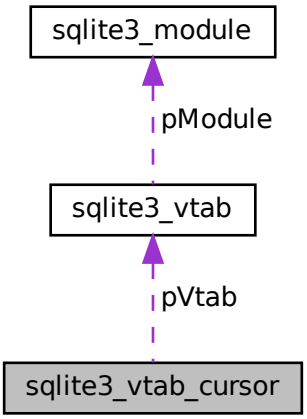
- const sqlite3_module * **pModule**
- int **nRef**
- char * **zErrMsg**

The documentation for this struct was generated from the following file:

- include/sqlite3.h

3.28 sqlite3_vtab_cursor Struct Reference

Collaboration diagram for sqlite3_vtab_cursor:



Public Attributes

- sqlite3_vtab * **pVtab**

The documentation for this struct was generated from the following file:

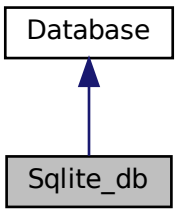
- include/sqlite3.h

3.29 Sqlite_db Class Reference

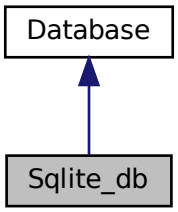
The Database class encapsulates management of SQLite databases.

`#include <sqlite_database.hpp>`

Inheritance diagram for Sqlite_db:



Collaboration diagram for Sqlite_db:



Public Member Functions

- Sqlite_db ()
  default constructor.
- Sqlite_db (string name, string uri, DatabaseType type=SQLite_db)
  Parameterized constructor.
- ~Sqlite_db ()
  destructor.
- virtual string **get_uri** ()
- virtual string **get_name** ()
- virtual DatabaseType **get_type** ()
- virtual int32_t **create** ()
- virtual int32_t **drop** ()

Additional Inherited Members

3.29.1 Detailed Description

The Database class encapsulates management of SQLite databases.

This class represents Sqlite databases

SQLiteDatabase abstract class provides the following services

- create – creates the database in the DB management systems
- drop – drops the database in the DB management systems

3.29.2 Constructor & Destructor Documentation

3.29.2.1 Sqlite_db()

```
Sqlite_db::Sqlite_db (
            string name,
            string uri,
            DatabaseType type = SQLite_db )
```

Parameterized constructor.

Parameters

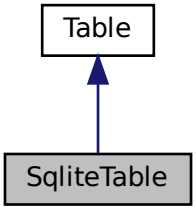
name	name of one of the database
uri	location of the database in the system
DatabaseType	this is the type of the database (default value is Sqlite_db)

The documentation for this class was generated from the following file:

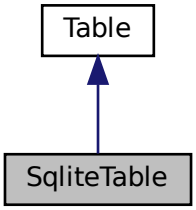
- include/sqlite_database.hpp

3.30 SqliteTable Class Reference

Inheritance diagram for SqliteTable:



Collaboration diagram for SqliteTable:



Public Member Functions

- SqliteTable ()
  default constructor.
- SqliteTable (string name)
  Parameterized constructor.
- ~SqliteTable ()
  destructor.

Additional Inherited Members

3.30.1 Constructor & Destructor Documentation

3.30.1.1 SqliteTable()

```
SqliteTable::SqliteTable (
            string name )
```

Parameterized constructor.

Parameters

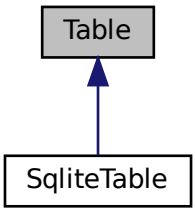
name	name of one of the database

The documentation for this class was generated from the following file:

- include/sqlite_database.hpp

3.31 Table Class Reference

Inheritance diagram for Table:



Public Member Functions

- virtual string **get_name** ()=0
- virtual int32_t **create** ()=0
- virtual int32_t **drop** ()=0

Protected Member Functions

- virtual string **get_description_sql** ()=0
- virtual string **get_create_stmt_sql** ()=0
- virtual string **get_drop_stmt_sql** ()=0

Protected Attributes

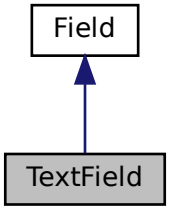
- string **name**
- vector< Field > **fields**

The documentation for this class was generated from the following file:

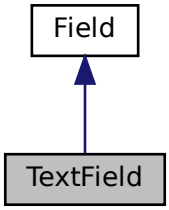
- include/database.hpp

3.32 TextField Class Reference

Inheritance diagram for TextField:



Collaboration diagram for TextField:



Public Member Functions

- TextField ()
  default constructor.
- TextField (string name, FieldType type=Text)
  Parameterized constructor.
- ~TextField ()
  destructor.

Additional Inherited Members

3.32.1 Constructor & Destructor Documentation

3.32.1.1 TextField()

```
TextField::TextField (
            string name,
            FieldType type = Text )
```

Parameterized constructor.

Parameters

name	name of one of the database
FieldType	this is the type of the database (default value is Text)

The documentation for this class was generated from the following file:

- include/sqlite_database.hpp

Index