LemonDB

Generated by Doxygen 1.8.13

Contents

1 LemonDB **1**

2 Hierarchical Index **5**

2.1 Class Hierarchy 5

3 Class Index **7**

3.1 Class List 7

4 Class Documentation **9**

4.1 AddQuery Class Reference 9

4.1.1 Detailed Description 10

4.2 AddTask Class Reference 10

4.2.1 Detailed Description 10

4.3 AnswerResult Class Reference 11

4.3.1 Detailed Description 11

4.4 BasicQueryBuilder Class Reference 11

4.4.1 Detailed Description 12

4.5 ComplexQuery Class Reference 12

4.5.1 Detailed Description 13

4.5.2 Member Function Documentation 13

4.5.2.1 evalCondition() 13

4.5.2.2 getCondition() 13

4.5.2.3 getOperands() 14

4.5.2.4 initCondition() 14

4.5.2.5 testKeyCondition() 14

4.5.3 Member Data Documentation . . . 15
4.5.3.1 condition . . . 15
4.5.3.2 operands . . . 15
4.6 ComplexQueryBuilder Class Reference . . . 15
4.6.1 Detailed Description . . . 16
4.7 ConflictingKey Struct Reference . . . 16
4.7.1 Detailed Description . . . 16
4.8 CopyTableDestQuery Class Reference . . . 16
4.8.1 Detailed Description . . . 17
4.9 CopyTableQuery Class Reference . . . 17
4.9.1 Detailed Description . . . 18
4.10 CountQuery Class Reference . . . 18
4.10.1 Detailed Description . . . 18
4.11 CountTask Class Reference . . . 19
4.11.1 Detailed Description . . . 19
4.12 Database Class Reference . . . 19
4.12.1 Detailed Description . . . 20
4.12.2 Member Function Documentation . . . 20
4.12.2.1 addQuery() . . . 20
4.12.2.2 addTask() . . . 20
4.12.2.3 completeQuery() . . . 20
4.12.2.4 ensureTable() . . . 21
4.13 DeleteQuery Class Reference . . . 21
4.13.1 Detailed Description . . . 22
4.14 DeleteTask Class Reference . . . 22
4.14.1 Detailed Description . . . 22
4.15 DropTableQuery Class Reference . . . 23
4.15.1 Detailed Description . . . 23
4.16 DumpTableQuery Class Reference . . . 23
4.16.1 Detailed Description . . . 24

4.17 DumpTableTask Class Reference . . . 24
4.17.1 Detailed Description . . . 24
4.18 DuplicatedTableName Struct Reference . . . 25
4.18.1 Detailed Description . . . 25
4.19 DuplicateQuery Class Reference . . . 25
4.19.1 Detailed Description . . . 26
4.20 DuplicateTask Class Reference . . . 26
4.20.1 Detailed Description . . . 26
4.21 ErrorMsgResult Class Reference . . . 27
4.21.1 Detailed Description . . . 27
4.22 FailedQueryBuilder Class Reference . . . 27
4.22.1 Detailed Description . . . 28
4.23 FailedQueryResult Class Reference . . . 28
4.23.1 Detailed Description . . . 28
4.24 IllFormedQuery Struct Reference . . . 29
4.24.1 Detailed Description . . . 29
4.25 IllFormedQueryCondition Struct Reference . . . 29
4.25.1 Detailed Description . . . 29
4.26 InsertQuery Class Reference . . . 30
4.26.1 Detailed Description . . . 30
4.27 InsertTask Class Reference . . . 30
4.27.1 Detailed Description . . . 31
4.28 Table::IteratorImpl< ObjType, DatumIterator > Class Template Reference . . . 31
4.28.1 Detailed Description . . . 32
4.29 ListTableQuery Class Reference . . . 32
4.29.1 Detailed Description . . . 32
4.30 LoadFromStreamException Struct Reference . . . 32
4.30.1 Detailed Description . . . 33
4.31 LoadTableQuery Class Reference . . . 33
4.31.1 Detailed Description . . . 33

4.32 LoadTableTask Class Reference . . . 34
4.32.1 Detailed Description . . . 34
4.33 MaxQuery Class Reference . . . 34
4.33.1 Detailed Description . . . 35
4.34 MaxTask Class Reference . . . 35
4.34.1 Detailed Description . . . 36
4.35 MinQuery Class Reference . . . 36
4.35.1 Detailed Description . . . 37
4.36 MinTask Class Reference . . . 37
4.36.1 Detailed Description . . . 37
4.37 MultipleKey Struct Reference . . . 38
4.37.1 Detailed Description . . . 38
4.38 NopQuery Class Reference . . . 38
4.38.1 Detailed Description . . . 38
4.39 NullQueryResult Class Reference . . . 39
4.39.1 Detailed Description . . . 39
4.40 Table::ObjectImpl< Iterator, VType > Class Template Reference . . . 39
4.40.1 Detailed Description . . . 40
4.41 PrintTableQuery Class Reference . . . 40
4.41.1 Detailed Description . . . 40
4.42 Query Class Reference . . . 41
4.42.1 Detailed Description . . . 41
4.42.2 Member Function Documentation . . . 42
4.42.2.1 getId() . . . 42
4.42.2.2 initId() . . . 42
4.43 QueryBuilder Class Reference . . . 42
4.43.1 Detailed Description . . . 43
4.44 QueryBuilderMatchFailed Class Reference . . . 43
4.44.1 Detailed Description . . . 43
4.45 QueryCondition Struct Reference . . . 44

4.45.1 Detailed Description . . . 44
4.46 QueryParser Class Reference . . . 44
4.46.1 Detailed Description . . . 44
4.47 QueryResult Class Reference . . . 44
4.47.1 Detailed Description . . . 45
4.48 QuitQuery Class Reference . . . 45
4.48.1 Detailed Description . . . 45
4.49 RecordCountResult Class Reference . . . 46
4.49.1 Detailed Description . . . 46
4.50 SelectQuery Class Reference . . . 46
4.50.1 Detailed Description . . . 47
4.51 SelectResult Class Reference . . . 47
4.51.1 Detailed Description . . . 47
4.52 SelectTask Class Reference . . . 48
4.52.1 Detailed Description . . . 48
4.53 SubQuery Class Reference . . . 48
4.53.1 Detailed Description . . . 49
4.54 SubTask Class Reference . . . 49
4.54.1 Detailed Description . . . 50
4.55 SuccessMsgResult Class Reference . . . 50
4.55.1 Detailed Description . . . 50
4.56 SuceededQueryResult Class Reference . . . 51
4.56.1 Detailed Description . . . 51
4.57 SumQuery Class Reference . . . 51
4.57.1 Detailed Description . . . 52
4.58 SumTask Class Reference . . . 52
4.58.1 Detailed Description . . . 53
4.59 SwapQuery Class Reference . . . 53
4.59.1 Detailed Description . . . 53
4.60 SwapTask Class Reference . . . 54

4.60.1 Detailed Description . . . 54
4.61 Table Class Reference . . . 54
4.61.1 Detailed Description . . . 56
4.61.2 Member Function Documentation . . . 56
4.61.2.1 begin() [1/2] . . . 56
4.61.2.2 begin() [2/2] . . . 56
4.61.2.3 clear() . . . 56
4.61.2.4 duplicate() . . . 57
4.61.2.5 empty() . . . 57
4.61.2.6 end() [1/2] . . . 57
4.61.2.7 end() [2/2] . . . 57
4.61.2.8 erase() . . . 57
4.61.2.9 eraseUnique() . . . 58
4.61.2.10 field() . . . 58
4.61.2.11 mergeData() . . . 58
4.61.2.12 move() . . . 58
4.61.2.13 name() . . . 59
4.61.2.14 operator[]() . . . 59
4.61.2.15 setName() . . . 59
4.61.2.16 size() . . . 60
4.61.2.17 swapData() . . . 60
4.61.3 Friends And Related Function Documentation . . . 60
4.61.3.1 operator<< . . . 60
4.62 TableFieldNotFound Struct Reference . . . 61
4.62.1 Detailed Description . . . 61
4.63 TableNameNotFound Struct Reference . . . 61
4.63.1 Detailed Description . . . 62
4.64 Task Class Reference . . . 62
4.64.1 Detailed Description . . . 63
4.64.2 Member Data Documentation . . . 63

4.64.2.1 counter . . . 63
4.65 TaskQuery Class Reference . . . 64
4.65.1 Detailed Description . . . 64
4.65.2 Member Function Documentation . . . 65
4.65.2.1 addIterationTask() . . . 65
4.65.2.2 addUniqueTask() . . . 65
4.65.2.3 complete() [1/2] . . . 65
4.65.2.4 complete() [2/2] . . . 65
4.65.2.5 start() . . . 66
4.65.3 Member Data Documentation . . . 66
4.65.3.1 taskComplete . . . 66
4.65.3.2 tasks . . . 66
4.65.3.3 tasksMutex . . . 66
4.65.3.4 tasksSize . . . 66
4.66 TokenizedQueryString Struct Reference . . . 67
4.66.1 Detailed Description . . . 67
4.67 TruncateTableQuery Class Reference . . . 67
4.67.1 Detailed Description . . . 67
4.68 UnableToOpenFile Struct Reference . . . 68
4.68.1 Detailed Description . . . 68
4.69 UpdateQuery Class Reference . . . 68
4.69.1 Detailed Description . . . 69
4.70 UpdateTask Class Reference . . . 69
4.70.1 Detailed Description . . . 69

Index 71

Chapter 1

LemonDB

Introduction

A simple multi-thread key-value database by Lemonion. Inc.

See more information in our official documentation HTML/PDF

Compilation

Debug

This version is used for debugging.

```
mkdir debug && cd debug
cmake ..  -DCMAKE_BUILD_TYPE=Debug
make lemondb
src/lemondb
```

Release

This version is used for performance test.

```
mkdir release && cd release
cmake ..  -DCMAKE_BUILD_TYPE=Release
make lemondb
src/lemondb
```

Testing

For a small test case, just use the files under `test` folder. Set the working directory as `test`, set the program argument as `test.query` or `test*.in`.

The test cases are too bug, so they are stored with Git LFS. See more information on `Git LFS pages`.

Once Git LFS extension is installed, you can download the test cases through cloning the submodule:

```
git submodule init
git submodule update
```

Set the working directory as `testcase/sample`, set the program argument as `*.query`, simply start debugging! (The loading query in all test files should be based on `testcase/sample` directory)

Documentation

The working flow of LemonDB is written by ourselves.

The class / function documentation is generated by `Doxygen`.

Design

- We design this program to make it can create 8 threads. We classify the queries using table and add them into queryqueue of corresponding table when we read them from the file. Once the table needed has been loaded completely, we will execute the queries as the order in queue and read query if the file hasn't ended at the same time.
- Queries in the queryqueue will be run in the parallel.
- Even in one query, for data queries that need searching and calculation such as SUB or SUM in a large table, we use a function addIterationTask to divide the table into several parts and search or calculate these parts simultaneously. Because the efficiency depends on the ratio between size of every part and size of the table, we did experiments and then find 10000 is a good size. For queries like TRUNCATE and INSERT, since they don't have to traverse the whole table, we don't add task for them.
- The query class will use a "combine" function to check whether all the tasks dispatched by one query are all finished and organize them in the order and show on the screen.
- When the user ask for quit, the quit query will check whether all tasks have already finished and and exit the program.

Performance Improvement

We use many tricks to improve performance:

- Since we use a vector to store all data in a table, we obtain the advantage of efficient random access. Meanwhile, deleting and inserting datum becomes less efficient inevitably. However, we use some tricks to handle this issue. Notice that the vector is unordered, so for INSERT query it can be trivially appended to the vector with O(1). Then for DELETE and DUPLICATE query, we use a temporary vector `dataNew`. When iterating through data, those won't be deleted will be moved to dataNew by `std::move`, which is extremely fast. Then we simply swap `data` and `dataNew`, clear `dataNew` for further use. For DUPLICATE, things are slightly different. We insert duplicated data into `dataNew`, and then we append `dataNew` to `data`. These iteration can be executed in parallel, so `dataNew` is, of course, protected by a mutex.
- Another great improvement is for those query with a condition 'KEY = someKey'. Making use of efficient random access, we keep a `keyMap` which stores index for given key. With this map, we can complete those query very efficiently, without iterating through data.
- For those query without given key, we also improve the speed of evaluating condition. This is done by computing the condition explicitly for a specific query (by `std::function`), and simply pass this function to it. By doing so, we don't need to repeatedly compare string, convert string to integer, even switch among operators. This can save huge amount of time, because originally every datum use one general evaluating function. Now we just need to compute it once per query.
- In some trivial cases atomic_int is used instead of having a mutex because it is much faster.

Problems Solved

Due to our sophisticated design, we ran into many problems. These are some of them:

- We have encountered many problems about the query queue. The problem of LOAD query is the most difficult one, because it doesn't specify a table name. In our design, every table has a query queue so that we can decide the order to execute them, following reader/writer pattern. But LOAD doesn't have it, so it's very difficult to deicide where to put it, because the file may even not exist when the query is parsed and put into some queue. DUMP query is the reason we concern about this issue, so we solve it by keeping a map from filename to tablename. With this map, LOAD can decide whether the file is (or will be) created by DUMP or should exist already.
- Another issue is COPYTABLE. This is the other query which can create a table, in which case it is responsible for starting the query queue to execute. And the problem is that COPYTABLE involves 2 table, so it should be pushed to both tables. And only when both query queues come to this query should it execute. This is done by keeping each other's pointer in it.
- Other problems such as mutex or deadlock are less encountered, because we consider carefully about implementation before we start to code.

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Database . . . 19
invalid_argument
  ConflictingKey . . . 16
  DuplicatedTableName . . . 25
  IllFormedQuery . . . 29
  IllFormedQueryCondition . . . 29
  MultipleKey . . . 38
  QueryBuilderMatchFailed . . . 43
  TableNameNotFound . . . 61
Table::IteratorImpl< ObjType, DatumIterator > . . . 31
Table::IteratorImpl< Object, decltype(data.begin())> . . . 31
Table::ObjectImpl< Iterator, VType > . . . 39
out_of_range
  TableFieldNotFound . . . 61
Query . . . 41
  ListTableQuery . . . 32
  NopQuery . . . 38
  PrintTableQuery . . . 40
  TaskQuery . . . 64
    ComplexQuery . . . 12
      AddQuery . . . 9
      CountQuery . . . 18
      DeleteQuery . . . 21
      DuplicateQuery . . . 25
      InsertQuery . . . 30
      MaxQuery . . . 34
      MinQuery . . . 36
      SelectQuery . . . 46
      SubQuery . . . 48
      SumQuery . . . 51
      SwapQuery . . . 53
      UpdateQuery . . . 68
    CopyTableDestQuery . . . 16
    CopyTableQuery . . . 17
    DropTableQuery . . . 23

DumpTableQuery . . . 23
LoadTableQuery . . . 33
QuitQuery . . . 45
TruncateTableQuery . . . 67
QueryBuilder . . . 42
BasicQueryBuilder . . . 11
ComplexQueryBuilder . . . 15
FailedQueryBuilder . . . 27
QueryCondition . . . 44
QueryParser . . . 44
QueryResult . . . 44
FailedQueryResult . . . 28
ErrorMsgResult . . . 27
SuceededQueryResult . . . 51
AnswerResult . . . 11
NullQueryResult . . . 39
RecordCountResult . . . 46
SelectResult . . . 47
SuccessMsgResult . . . 50
runtime_error
LoadFromStreamException . . . 32
UnableToOpenFile . . . 68
Table . . . 54
Task . . . 62
AddTask . . . 10
CountTask . . . 19
DeleteTask . . . 22
DumpTableTask . . . 24
DuplicateTask . . . 26
InsertTask . . . 30
LoadTableTask . . . 34
MaxTask . . . 35
MinTask . . . 37
SelectTask . . . 48
SubTask . . . 49
SumTask . . . 52
SwapTask . . . 54
UpdateTask . . . 69
TokenizedQueryString . . . 67

Chapter 3

Class Index

3.1 Class List

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

AddQuery . . . 9
AddTask . . . 10
AnswerResult . . . 11
BasicQueryBuilder . . . 11
ComplexQuery . . . 12
ComplexQueryBuilder . . . 15
ConflictingKey . . . 16
CopyTableDestQuery . . . 16
CopyTableQuery . . . 17
CountQuery . . . 18
CountTask . . . 19
Database . . . 19
DeleteQuery . . . 21
DeleteTask . . . 22
DropTableQuery . . . 23
DumpTableQuery . . . 23
DumpTableTask . . . 24
DuplicatedTableName . . . 25
DuplicateQuery . . . 25
DuplicateTask . . . 26
ErrorMsgResult . . . 27
FailedQueryBuilder . . . 27
FailedQueryResult . . . 28
IllFormedQuery . . . 29
IllFormedQueryCondition . . . 29
InsertQuery . . . 30
InsertTask . . . 30
Table::IteratorImpl< ObjType, DatumIterator > . . . 31
ListTableQuery . . . 32
LoadFromStreamException . . . 32
LoadTableQuery . . . 33
LoadTableTask . . . 34
MaxQuery . . . 34
MaxTask . . . 35
MinQuery . . . 36

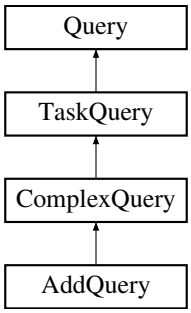
MinTask . . . 37
MultipleKey . . . 38
NopQuery . . . 38
NullQueryResult . . . 39
Table::ObjectImpl< Iterator, VType > . . . 39
PrintTableQuery . . . 40
Query . . . 41
QueryBuilder . . . 42
QueryBuilderMatchFailed . . . 43
QueryCondition . . . 44
QueryParser . . . 44
QueryResult . . . 44
QuitQuery . . . 45
RecordCountResult . . . 46
SelectQuery . . . 46
SelectResult . . . 47
SelectTask . . . 48
SubQuery . . . 48
SubTask . . . 49
SuccessMsgResult . . . 50
SuceededQueryResult . . . 51
SumQuery . . . 51
SumTask . . . 52
SwapQuery . . . 53
SwapTask . . . 54
Table . . . 54
TableFieldNotFound . . . 61
TableNameNotFound . . . 61
Task . . . 62
TaskQuery . . . 64
TokenizedQueryString . . . 67
TruncateTableQuery . . . 67
UnableToOpenFile . . . 68
UpdateQuery . . . 68
UpdateTask . . . 69

Chapter 4

Class Documentation

4.1 AddQuery Class Reference

Inheritance diagram for AddQuery:



Public Member Functions

- **LEMONDB_QUERY_WRITER** (true)
- QueryResult::Ptr **execute** () override
- std::string **toString** () override
- QueryResult::Ptr **combine** (int taskComplete) override

Protected Member Functions

- **LEMONDB_TASK_PTR_DEF** (AddTask)

Friends

- class **AddTask**

Additional Inherited Members

4.1.1 Detailed Description

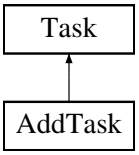
Definition at line 9 of file add_query.h.

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

- src/query/data/add_query.h
- src/query/data/add_query.cpp

4.2 AddTask Class Reference

Inheritance diagram for AddTask:



Public Member Functions

- void **execute** () override

Protected Member Functions

- **LEMONDB_QUERY_PTR** (AddQuery)

Friends

- class **AddQuery**

Additional Inherited Members

4.2.1 Detailed Description

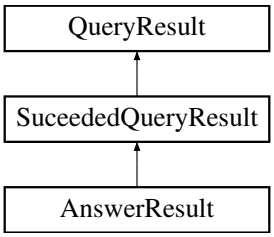
Definition at line 23 of file add_query.h.

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

- src/query/data/add_query.h
- src/query/data/add_query.cpp

4.3 AnswerResult Class Reference

Inheritance diagram for AnswerResult:



Public Member Functions

- **AnswerResult** (std::vector< int > &&answer)
- **AnswerResult** (int answer)
- std::string **toString** () override

Additional Inherited Members

4.3.1 Detailed Description

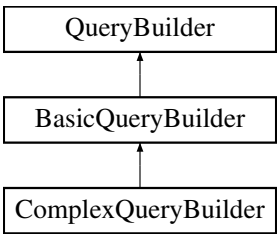
Definition at line 108 of file query_results.h.

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

- src/query_results.h

4.4 BasicQueryBuilder Class Reference

Inheritance diagram for BasicQueryBuilder:



Public Member Functions

- void **setNext** (Ptr &&builder) override
- Query::Ptr **tryExtractQuery** (TokenizedQueryString &query) override
- void **clear** () override

Protected Attributes

- QueryBuilder::Ptr **nextBuilder**

Additional Inherited Members

4.4.1 Detailed Description

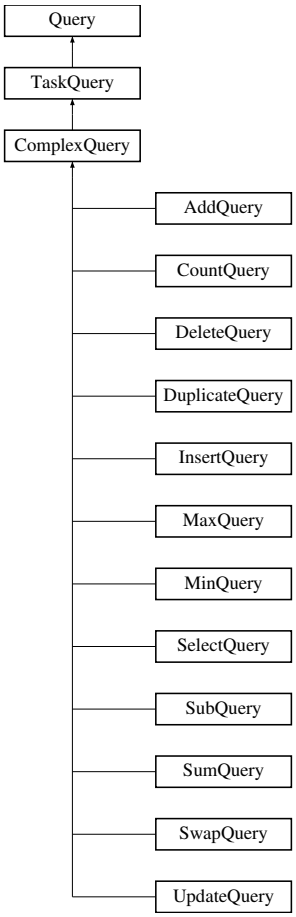
Definition at line 41 of file query_builders.h.

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

- src/query_builders.h

4.5 ComplexQuery Class Reference

Inheritance diagram for ComplexQuery:



Public Types

- typedef std::unique_ptr< ComplexQuery > **Ptr**

Public Member Functions

- std::pair< std::string, bool > initCondition (const Table &table)
- bool evalCondition (const Table::Object &object)
- bool testKeyCondition (Table &table, std::function< void(bool, Table::Object::Ptr &&)> function)
- **ComplexQuery** (std::string targetTable, std::vector< std::string > operands, std::vector< QueryCondition > condition)
- const std::vector< std::string > & getOperands () const
- const std::vector< QueryCondition > & getCondition ()

Protected Attributes

- std::vector< std::string > operands
- std::vector< QueryCondition > condition

4.5.1 Detailed Description

Definition at line 100 of file query.h.

4.5.2 Member Function Documentation

4.5.2.1 evalCondition()

```
bool ComplexQuery::evalCondition (
            const Table::Object & object )
```

skip the evaluation of KEY (which should be done after initConditionFast is called)

Parameters

conditions	
object	

Returns

Definition at line 101 of file query.cpp.

4.5.2.2 getCondition()

```
const std::vector<QueryCondition>& ComplexQuery::getCondition ( )  [inline]
```

Get condition in the query, seems no use now

Definition at line 150 of file query.h.

4.5.2.3 getOperands()

```
const std::vector<std::string>& ComplexQuery::getOperands ( ) const  [inline]
```

Get operands in the query

Definition at line 147 of file query.h.

4.5.2.4 initCondition()

```
std::pair< std::string, bool > ComplexQuery::initCondition (
            const Table & table )
```

init a fast condition according to the table note that the condition is only effective if the table fields are not changed

Parameters

table	
conditions	

Returns

a pair of the key and a flag if flag is false, the condition is always false in this situation, the condition may not be fully initialized to save time

Definition at line 45 of file query.cpp.

4.5.2.5 testKeyCondition()

```
bool ComplexQuery::testKeyCondition (
            Table & table,
            std::function< void(bool, Table::Object::Ptr &&)> function )
```

This function seems have small effect and causes somme bugs so it is not used actually

Parameters

table	
function	

Returns

Definition at line 113 of file query.cpp.

4.5.3 Member Data Documentation

4.5.3.1 condition

```
std::vector<QueryCondition> ComplexQuery::condition  [protected]
```

The function used in where clause

Definition at line 105 of file query.h.

4.5.3.2 operands

```
std::vector<std::string> ComplexQuery::operands  [protected]
```

The field names in the first ()

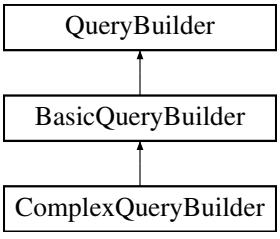
Definition at line 103 of file query.h.

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

- src/query/query.h
- src/query/query.cpp

4.6 ComplexQueryBuilder Class Reference

Inheritance diagram for ComplexQueryBuilder:



Public Member Functions

- void **clear** () override
- Query::Ptr **tryExtractQuery** (TokenizedQueryString &query) override

Protected Member Functions

- virtual void **parseToken** (TokenizedQueryString &query)

Protected Attributes

- std::string **targetTable**
- std::vector< std::string > **operandToken**
- std::vector< QueryCondition > **conditionToken**

Additional Inherited Members

4.6.1 Detailed Description

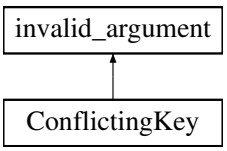
Definition at line 60 of file query_builders.h.

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

- src/query_builders.h
- src/query_builders.cpp

4.7 ConflictingKey Struct Reference

Inheritance diagram for ConflictingKey:



Public Member Functions

- **ConflictingKey** (const std::string &str)

4.7.1 Detailed Description

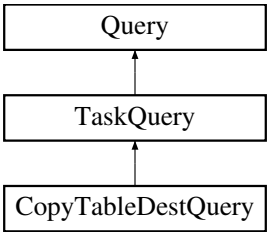
Definition at line 26 of file uexception.h.

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

- src/uexception.h

4.8 CopyTableDestQuery Class Reference

Inheritance diagram for CopyTableDestQuery:



Public Member Functions

- **LEMONDB_QUERY_WRITER** (true)
- **LEMONDB_QUERY_INSTANT** (true)
- **CopyTableDestQuery** (std::string table, CopyTableQuery *srcQuery)
- QueryResult::Ptr **execute** () override
- std::string **toString** () override

Friends

- class **CopyTableQuery**

Additional Inherited Members

4.8.1 Detailed Description

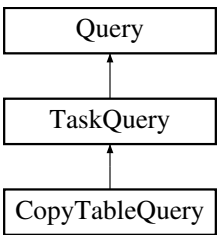
Definition at line 26 of file copy_table_query.h.

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

- src/query/management/copy_table_query.h
- src/query/management/copy_table_query.cpp

4.9 CopyTableQuery Class Reference

Inheritance diagram for CopyTableQuery:



Public Member Functions

- **LEMONDB_QUERY_WRITER** (false)
- **CopyTableQuery** (std::string table, std::string newTable)
- QueryResult::Ptr **execute** () override
- std::string **toString** () override
- Query::Ptr **createDestQuery** ()

Additional Inherited Members

4.9.1 Detailed Description

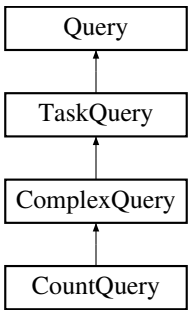
Definition at line 13 of file copy_table_query.h.

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

- src/query/management/copy_table_query.h
- src/query/management/copy_table_query.cpp

4.10 CountQuery Class Reference

Inheritance diagram for CountQuery:



Public Member Functions

- **LEMONDB_QUERY_WRITER** (false)
- QueryResult::Ptr **execute** () override
- std::string **toString** () override
- QueryResult::Ptr **combine** (int taskComplete) override

Additional Inherited Members

4.10.1 Detailed Description

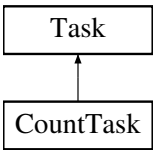
Definition at line 7 of file count_query.h.

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

- src/query/data/count_query.h
- src/query/data/count_query.cpp

4.11 CountTask Class Reference

Inheritance diagram for CountTask:



Public Member Functions

- void **execute** () override

Protected Member Functions

- **LEMONDB_QUERY_PTR** (CountQuery)

Additional Inherited Members

4.11.1 Detailed Description

Definition at line 17 of file count_query.h.

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

- src/query/data/count_query.h
- src/query/data/count_query.cpp

4.12 Database Class Reference

Public Member Functions

- void **registerTable** (Table::Ptr &&table)
- Table & ensureTable (const std::string &tableName)
- void **dropTable** (std::string tableName)
- void **printAllTable** ()
- Table & **operator[ ]** (std::string tableName)
- const Table & **operator[ ]** (std::string tableName) const
- Database & **operator=** (const Database &)=delete
- Database & **operator=** (Database &&)=delete
- **Database** (const Database &)=delete
- **Database** (Database &&)=delete
- void **updateFileTableName** (const std::string &fileName, const std::string &tableName)
- std::string **getFileTableName** (const std::string &fileName)
- void addQuery (Query::Ptr &&query)
- void addTask (Task *task)
- void **addResult** (Query *query, QueryResult::Ptr &&result)
- void completeQuery ()
- void **endQuery** ()
- bool **isEnd** () const
- void **joinThreads** ()

Static Public Member Functions

- static Database & **getInstance** ()

4.12.1 Detailed Description

Definition at line 16 of file db.h.

4.12.2 Member Function Documentation

4.12.2.1 addQuery()

```
void Database::addQuery (
            Query::Ptr && query )
```

Add a parsed query after reading it dispatch the query according to its target table

Parameters

query	

Definition at line 117 of file db.cpp.

4.12.2.2 addTask()

```
void Database::addTask (
            Task * task )
```

Add a generated task after a query has been executed by a table idle working threads are waiting for the task

Parameters

task	

Definition at line 146 of file db.cpp.

4.12.2.3 completeQuery()

```
void Database::completeQuery ( )
```

try to output the query result in order

Definition at line 165 of file db.cpp.

4.12.2.4 ensureTable()

```
Table & Database::ensureTable (
            const std::string & tableName )
```

get the table if it already exists create a table if tableName not found use tablesMutex, call it when needed if table↩ Name must exist, use operator[]

Parameters

tableName	

Returns

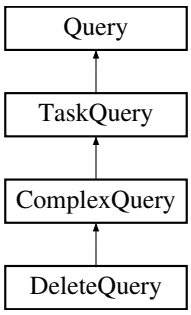
Definition at line 43 of file db.cpp.

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

- src/db/db.h
- src/db/db.cpp

4.13 DeleteQuery Class Reference

Inheritance diagram for DeleteQuery:



Public Member Functions

- **LEMONDB_QUERY_WRITER** (true)
- QueryResult::Ptr **execute** () override
- std::string **toString** () override
- QueryResult::Ptr **combine** (int taskComplete) override

Friends

- class **DeleteTask**

Additional Inherited Members

4.13.1 Detailed Description

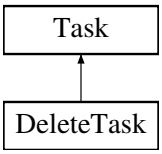
Definition at line 11 of file delete_query.h.

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

- src/query/data/delete_query.h
- src/query/data/delete_query.cpp

4.14 DeleteTask Class Reference

Inheritance diagram for DeleteTask:



Public Member Functions

- void **execute** () override

Protected Member Functions

- **LEMONDB_QUERY_PTR** (DeleteQuery)

Additional Inherited Members

4.14.1 Detailed Description

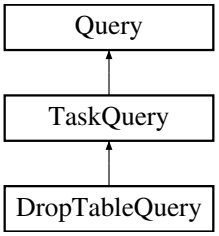
Definition at line 22 of file delete_query.h.

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

- src/query/data/delete_query.h
- src/query/data/delete_query.cpp

4.15 DropTableQuery Class Reference

Inheritance diagram for DropTableQuery:



Public Member Functions

- QueryResult::Ptr **execute** () override
- std::string **toString** () override

Additional Inherited Members

4.15.1 Detailed Description

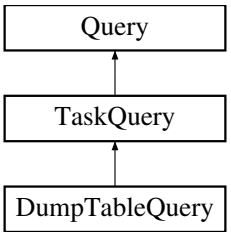
Definition at line 11 of file drop_query.h.

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

- src/query/management/drop_query.h
- src/query/management/drop_query.cpp

4.16 DumpTableQuery Class Reference

Inheritance diagram for DumpTableQuery:



Public Member Functions

- **DumpTableQuery** (std::string table, std::string filename)
- QueryResult::Ptr **execute** () override
- std::string **toString** () override
- QueryResult::Ptr **combine** (int taskComplete) override

Friends

- class **DumpTableTask**

Additional Inherited Members

4.16.1 Detailed Description

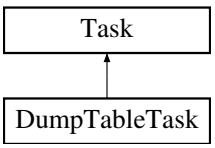
Definition at line 11 of file dump_query.h.

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

- src/query/management/dump_query.h
- src/query/management/dump_query.cpp

4.17 DumpTableTask Class Reference

Inheritance diagram for DumpTableTask:



Public Member Functions

- void **execute** () override

Protected Member Functions

- **LEMONDB_QUERY_PTR** (DumpTableQuery)

Additional Inherited Members

4.17.1 Detailed Description

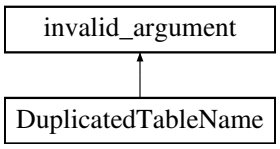
Definition at line 24 of file dump_query.h.

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

- src/query/management/dump_query.h
- src/query/management/dump_query.cpp

4.18 DuplicatedTableName Struct Reference

Inheritance diagram for DuplicatedTableName:



Public Member Functions

- **DuplicatedTableName** (const std::string &str)

4.18.1 Detailed Description

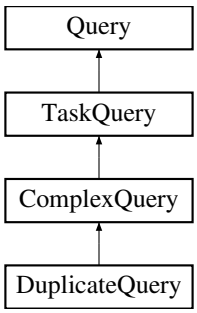
Definition at line 16 of file uexception.h.

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

- src/uexception.h

4.19 DuplicateQuery Class Reference

Inheritance diagram for DuplicateQuery:



Public Member Functions

- **LEMONDB_QUERY_WRITER** (true)
- QueryResult::Ptr **execute** () override
- std::string **toString** () override
- QueryResult::Ptr **combine** (int taskComplete) override

Friends

- class **DuplicateTask**

Additional Inherited Members

4.19.1 Detailed Description

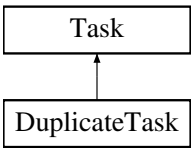
Definition at line 11 of file duplicate_query.h.

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

- src/query/data/duplicate_query.h
- src/query/data/duplicate_query.cpp

4.20 DuplicateTask Class Reference

Inheritance diagram for DuplicateTask:



Public Member Functions

- void **execute** () override

Protected Member Functions

- **LEMONDB_QUERY_PTR** (DuplicateQuery)

Additional Inherited Members

4.20.1 Detailed Description

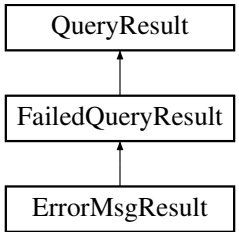
Definition at line 22 of file duplicate_query.h.

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

- src/query/data/duplicate_query.h
- src/query/data/duplicate_query.cpp

4.21 ErrorMsgResult Class Reference

Inheritance diagram for ErrorMsgResult:



Public Member Functions

- **ErrorMsgResult** (const char ∗qname, const std::string &msg)
- **ErrorMsgResult** (const char ∗qname, const char ∗table, const std::string &msg)
- std::string **toString** () override

Additional Inherited Members

4.21.1 Detailed Description

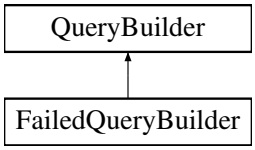
Definition at line 40 of file query_results.h.

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

- src/query_results.h

4.22 FailedQueryBuilder Class Reference

Inheritance diagram for FailedQueryBuilder:



Public Member Functions

- Query::Ptr **tryExtractQuery** (TokenizedQueryString &q) final
- void **setNext** (QueryBuilder::Ptr &&builder) final
- void **clear** () override

Static Public Member Functions

- static QueryBuilder::Ptr **getDefault** ()

Additional Inherited Members

4.22.1 Detailed Description

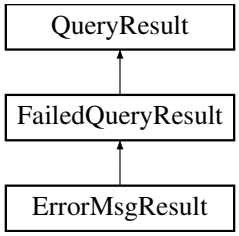
Definition at line 24 of file query_builders.h.

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

- src/query_builders.h

4.23 FailedQueryResult Class Reference

Inheritance diagram for FailedQueryResult:



Public Member Functions

- bool **success** () override

Additional Inherited Members

4.23.1 Detailed Description

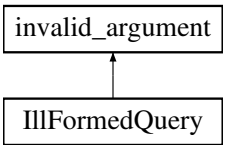
Definition at line 22 of file query_results.h.

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

- src/query_results.h

4.24 IllFormedQuery Struct Reference

Inheritance diagram for IllFormedQuery:



Public Member Functions

- **IllFormedQuery** (const std::string &str)

4.24.1 Detailed Description

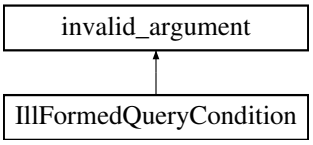
Definition at line 46 of file uexception.h.

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

- src/uexception.h

4.25 IllFormedQueryCondition Struct Reference

Inheritance diagram for IllFormedQueryCondition:



Public Member Functions

- **IllFormedQueryCondition** (const std::string &str)

4.25.1 Detailed Description

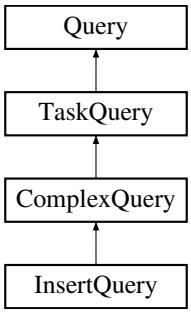
Definition at line 51 of file uexception.h.

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

- src/uexception.h

4.26 InsertQuery Class Reference

Inheritance diagram for InsertQuery:



Public Member Functions

- **LEMONDB_QUERY_WRITER** (true)
- QueryResult::Ptr **execute** () override
- std::string **toString** () override
- QueryResult::Ptr **combine** (int taskComplete) override

Friends

- class **InsertTask**

Additional Inherited Members

4.26.1 Detailed Description

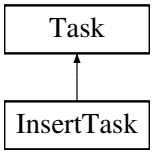
Definition at line 7 of file insert_query.h.

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

- src/query/data/insert_query.h
- src/query/data/insert_query.cpp

4.27 InsertTask Class Reference

Inheritance diagram for InsertTask:



Public Member Functions

- void **execute** () override

Protected Member Functions

- **LEMONDB_QUERY_PTR** (InsertQuery)

Additional Inherited Members

4.27.1 Detailed Description

Definition at line 18 of file insert_query.h.

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

- src/query/data/insert_query.h
- src/query/data/insert_query.cpp

4.28 Table::IteratorImpl< ObjType, DatumIterator > Class Template Reference

Public Member Functions

- **IteratorImpl** (DatumIterator datumIt, const Table *t)
- **IteratorImpl** (const IteratorImpl &)=default
- **IteratorImpl** (IteratorImpl &&) noexcept=default
- IteratorImpl & **operator=** (const IteratorImpl &)=default
- IteratorImpl & **operator=** (IteratorImpl &&) noexcept=default
- pointer **operator->** ()
- reference **operator*** ()
- IteratorImpl & **operator++** ()
- IteratorImpl & **operator--** ()
- IteratorImpl **operator++** (int)
- IteratorImpl **operator--** (int)
- bool **operator==** (const IteratorImpl &other)
- bool **operator!=** (const IteratorImpl &other)
- bool **operator<=** (const IteratorImpl &other)
- bool **operator>=** (const IteratorImpl &other)
- bool **operator<** (const IteratorImpl &other)
- bool **operator>** (const IteratorImpl &other)
- IteratorImpl **operator+** (int n)
- IteratorImpl **operator-** (int n)
- IteratorImpl & **operator+=** (int n)
- IteratorImpl & **operator-=** (int n)

Friends

- class **Table**

4.28.1 Detailed Description

template<typename ObjType, typename DatumIterator>
class Table::IteratorImpl< ObjType, DatumIterator >

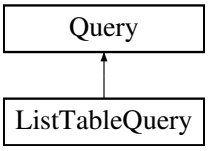
Definition at line 183 of file db_table.h.

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

- src/db/db_table.h

4.29 ListTableQuery Class Reference

Inheritance diagram for ListTableQuery:



Public Member Functions

- QueryResult::Ptr **execute** () override
- std::string **toString** () override

Additional Inherited Members

4.29.1 Detailed Description

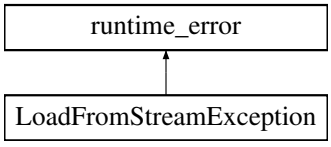
Definition at line 7 of file management_query.h.

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

- src/management_query.h
- src/management_query.cpp

4.30 LoadFromStreamException Struct Reference

Inheritance diagram for LoadFromStreamException:



Public Member Functions

- **LoadFromStreamException** (const std::string &str)

4.30.1 Detailed Description

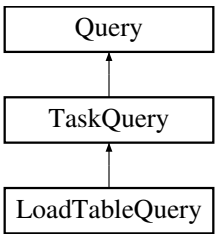
Definition at line 41 of file uexception.h.

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

- src/uexception.h

4.31 LoadTableQuery Class Reference

Inheritance diagram for LoadTableQuery:



Public Member Functions

- **LEMONDB_QUERY_WRITER** (true)
- **LEMONDB_QUERY_INSTANT** (true)
- **LoadTableQuery** (std::string table, std::string fileName)
- QueryResult::Ptr **execute** () override
- std::string **toString** () override
- QueryResult::Ptr **combine** (int taskComplete) override

Friends

- class **LoadTableTask**

Additional Inherited Members

4.31.1 Detailed Description

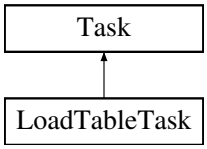
Definition at line 11 of file load_table_query.h.

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

- src/query/management/load_table_query.h
- src/query/management/load_table_query.cpp

4.32 LoadTableTask Class Reference

Inheritance diagram for LoadTableTask:



Public Member Functions

- void **execute** () override

Protected Member Functions

- **LEMONDB_QUERY_PTR** (LoadTableQuery)

Additional Inherited Members

4.32.1 Detailed Description

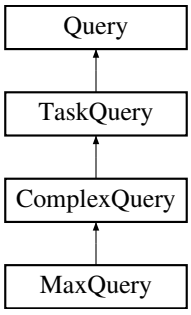
Definition at line 26 of file load_table_query.h.

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

- src/query/management/load_table_query.h
- src/query/management/load_table_query.cpp

4.33 MaxQuery Class Reference

Inheritance diagram for MaxQuery:



Public Member Functions

- **LEMONDB_QUERY_WRITER** (false)
- QueryResult::Ptr **execute** () override
- std::string **toString** () override
- QueryResult::Ptr **combine** (int taskComplete) override

Protected Member Functions

- **LEMONDB_TASK_PTR_DEF** (MaxTask)

Friends

- class **MaxTask**

Additional Inherited Members

4.33.1 Detailed Description

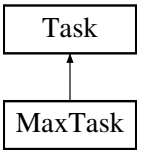
Definition at line 9 of file max_query.h.

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

- src/query/data/max_query.h
- src/query/data/max_query.cpp

4.34 MaxTask Class Reference

Inheritance diagram for MaxTask:



Public Member Functions

- void **execute** () override

Protected Member Functions

- **LEMONDB_QUERY_PTR** (MaxQuery)

Friends

- class **MaxQuery**

Additional Inherited Members

4.34.1 Detailed Description

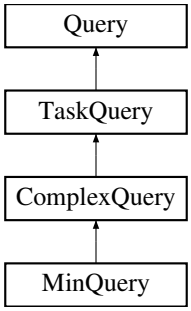
Definition at line 23 of file max_query.h.

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

- src/query/data/max_query.h
- src/query/data/max_query.cpp

4.35 MinQuery Class Reference

Inheritance diagram for MinQuery:



Public Member Functions

- **LEMONDB_QUERY_WRITER** (false)
- QueryResult::Ptr **execute** () override
- std::string **toString** () override
- QueryResult::Ptr **combine** (int taskComplete) override

Protected Member Functions

- **LEMONDB_TASK_PTR_DEF** (MinTask)

Friends

- class **MinTask**

Additional Inherited Members

4.35.1 Detailed Description

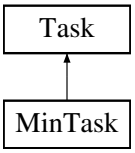
Definition at line 9 of file min_query.h.

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

- src/query/data/min_query.h
- src/query/data/min_query.cpp

4.36 MinTask Class Reference

Inheritance diagram for MinTask:



Public Member Functions

- void **execute** () override

Protected Member Functions

- **LEMONDB_QUERY_PTR** (MinQuery)

Friends

- class **MinQuery**

Additional Inherited Members

4.36.1 Detailed Description

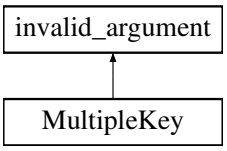
Definition at line 23 of file min_query.h.

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

- src/query/data/min_query.h
- src/query/data/min_query.cpp

4.37 MultipleKey Struct Reference

Inheritance diagram for MultipleKey:



Public Member Functions

- **MultipleKey** (const std::string &str)

4.37.1 Detailed Description

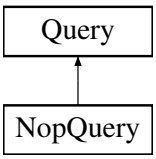
Definition at line 31 of file uexception.h.

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

- src/uexception.h

4.38 NopQuery Class Reference

Inheritance diagram for NopQuery:



Public Member Functions

- QueryResult::Ptr **execute** () override
- std::string **toString** () override

Additional Inherited Members

4.38.1 Detailed Description

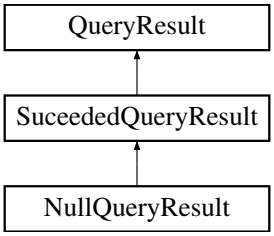
Definition at line 22 of file query.h.

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

- src/query/query.h

4.39 NullQueryResult Class Reference

Inheritance diagram for NullQueryResult:



Public Member Functions

- std::string **toString** () override

Additional Inherited Members

4.39.1 Detailed Description

Definition at line 33 of file query_results.h.

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

- src/query_results.h

4.40 Table::ObjectImpl< Iterator, VType > Class Template Reference

`#include <db_table.h>`

Public Types

- typedef std::unique_ptr< ObjectImpl > **Ptr**

Public Member Functions

- **ObjectImpl** (Iterator datumIt, const Table *t)
- **ObjectImpl** (const ObjectImpl &)=default
- **ObjectImpl** (ObjectImpl &&) noexcept=default
- ObjectImpl & **operator=** (const ObjectImpl &)=default
- ObjectImpl & **operator=** (ObjectImpl &&) noexcept=default
- KeyType **key** () const
- void **setKey** (KeyType key)
- VType & **operator[ ]** (const FieldNameType &field) const
- VType & **operator[ ]** (const FieldIndex &index) const
- VType & **get** (const FieldNameType &field) const
- VType & **get** (const FieldIndex &index) const

Friends

- class **Table**

4.40.1 Detailed Description

template<class Iterator, class VType>
class Table::ObjectImpl< Iterator, VType >

A proxy class that provides abstraction on internal Implementation. Allows independent variation on the Representation for a table object

Template Parameters

Iterator	
VType	

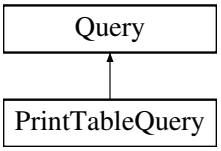
Definition at line 122 of file db_table.h.

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

- src/db/db_table.h

4.41 PrintTableQuery Class Reference

Inheritance diagram for PrintTableQuery:



Public Member Functions

- **PrintTableQuery** (std::string table)
- QueryResult::Ptr **execute** () override
- std::string **toString** () override

Additional Inherited Members

4.41.1 Detailed Description

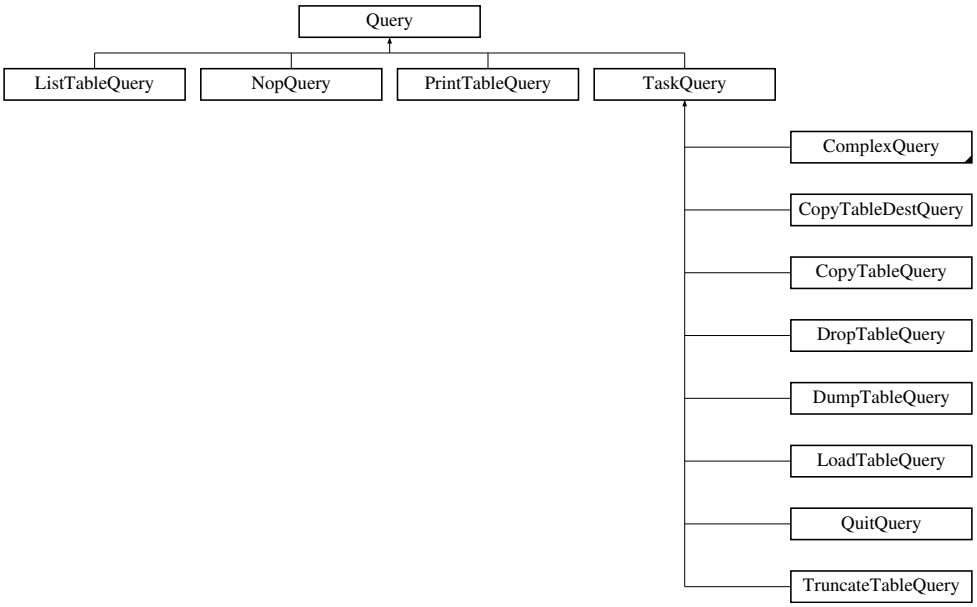
Definition at line 15 of file management_query.h.

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

- src/management_query.h
- src/management_query.cpp

4.42 Query Class Reference

Inheritance diagram for Query:



Public Types

- typedef std::unique_ptr< Query > **Ptr**

Public Member Functions

- virtual QueryResult::Ptr **execute** ()=0
- virtual std::string **toString** ()=0
- virtual QueryResult::Ptr **combine** (int taskComplete)
- virtual bool **isWriter** () const =0
- virtual bool **isInstant** () const
- const std::string & **getTableName** ()
- int getId () const
- int initId (int id)

Protected Attributes

- std::string **targetTable**
- int **id** = -1

4.42.1 Detailed Description

Definition at line 23 of file query_base.h.

4.42.2 Member Function Documentation

4.42.2.1 getId()

```
int Query::getId ( ) const  [inline]
```

get the unique id of this query

Returns

Definition at line 46 of file query_base.h.

4.42.2.2 initId()

```
int Query::initId (
            int id )  [inline]
```

will only work when first init the query id

Parameters

id	

Returns

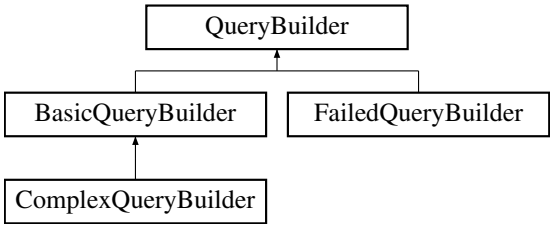
Definition at line 53 of file query_base.h.

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

- src/query/query_base.h

4.43 QueryBuilder Class Reference

Inheritance diagram for QueryBuilder:



Public Types

- typedef std::unique_ptr< QueryBuilder > **Ptr**

Public Member Functions

- virtual Query::Ptr **tryExtractQuery** (TokenizedQueryString &queryString)=0
- virtual void **setNext** (Ptr &&builder)=0
- virtual void **clear** ()=0

4.43.1 Detailed Description

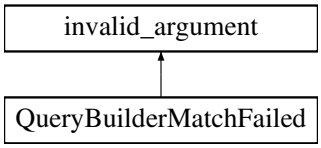
Definition at line 15 of file query_parser.h.

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

- src/query_parser.h

4.44 QueryBuilderMatchFailed Class Reference

Inheritance diagram for QueryBuilderMatchFailed:



Public Member Functions

- **QueryBuilderMatchFailed** (const std::string &qString)

4.44.1 Detailed Description

Definition at line 56 of file uexception.h.

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

- src/uexception.h

4.45 QueryCondition Struct Reference

Public Attributes

- std::string **field**
- size_t **fieldId**
- std::string **op**
- std::function< bool(const Table::ValueType &, const Table::ValueType &)> **comp**
- std::string **value**
- Table::ValueType **valueParsed**

4.45.1 Detailed Description

Definition at line 13 of file query.h.

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

- src/query/query.h

4.46 QueryParser Class Reference

Public Member Functions

- Query::Ptr **parseQuery** (std::string queryString)
- void **registerQueryBuilder** (QueryBuilder::Ptr &&qBuilder)

4.46.1 Detailed Description

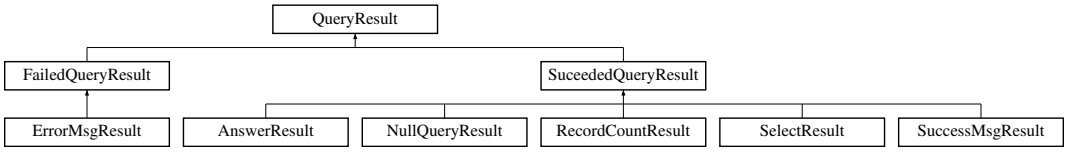
Definition at line 27 of file query_parser.h.

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

- src/query_parser.h
- src/query_parser.cpp

4.47 QueryResult Class Reference

Inheritance diagram for QueryResult:



Public Types

- typedef std::unique_ptr< QueryResult > **Ptr**

Public Member Functions

- virtual bool **success** ()=0
- virtual std::string **toString** ()=0

4.47.1 Detailed Description

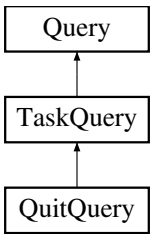
Definition at line 11 of file query_results.h.

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

- src/query_results.h

4.48 QuitQuery Class Reference

Inheritance diagram for QuitQuery:



Public Member Functions

- QueryResult::Ptr **execute** () override
- std::string **toString** () override

Additional Inherited Members

4.48.1 Detailed Description

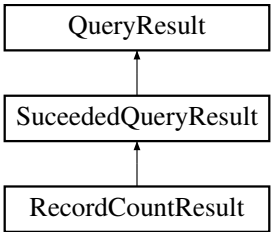
Definition at line 10 of file quit_query.h.

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

- src/query/management/quit_query.h
- src/query/management/quit_query.cpp

4.49 RecordCountResult Class Reference

Inheritance diagram for RecordCountResult:



Public Member Functions

- **RecordCountResult** (int count)
- std::string **toString** () override

Additional Inherited Members

4.49.1 Detailed Description

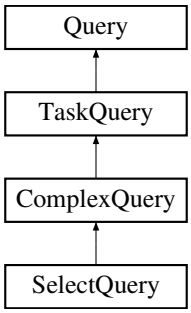
Definition at line 98 of file query_results.h.

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

- src/query_results.h

4.50 SelectQuery Class Reference

Inheritance diagram for SelectQuery:



Public Member Functions

- **LEMONDB_QUERY_WRITER** (false)
- QueryResult::Ptr **execute** () override
- std::string **toString** () override
- QueryResult::Ptr **combine** (int taskComplete) override

Protected Member Functions

- **LEMONDB_TASK_PTR_DEF** (SelectTask)

Friends

- class **SelectTask**

Additional Inherited Members

4.50.1 Detailed Description

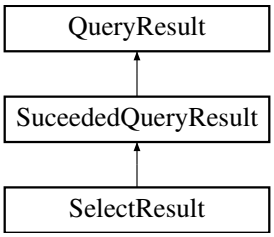
Definition at line 13 of file select_query.h.

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

- src/query/data/select_query.h
- src/query/data/select_query.cpp

4.51 SelectResult Class Reference

Inheritance diagram for SelectResult:



Public Member Functions

- **SelectResult** (std::vector< std::pair< std::string, std::vector< int > > > &&results)
- std::string **toString** () override

Additional Inherited Members

4.51.1 Detailed Description

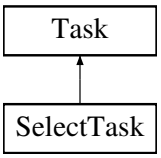
Definition at line 124 of file query_results.h.

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

- src/query_results.h

4.52 SelectTask Class Reference

Inheritance diagram for SelectTask:



Public Member Functions

- void **execute** () override

Protected Member Functions

- **LEMONDB_QUERY_PTR** (SelectQuery)

Friends

- class **SelectQuery**

Additional Inherited Members

4.52.1 Detailed Description

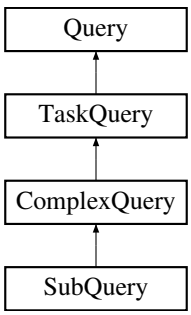
Definition at line 27 of file select_query.h.

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

- src/query/data/select_query.h
- src/query/data/select_query.cpp

4.53 SubQuery Class Reference

Inheritance diagram for SubQuery:



Public Member Functions

- **LEMONDB_QUERY_WRITER** (true)
- QueryResult::Ptr **execute** () override
- std::string **toString** () override
- QueryResult::Ptr **combine** (int taskComplete) override

Protected Member Functions

- **LEMONDB_TASK_PTR_DEF** (SubTask)

Friends

- class **SubTask**

Additional Inherited Members

4.53.1 Detailed Description

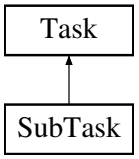
Definition at line 9 of file sub_query.h.

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

- src/query/data/sub_query.h
- src/query/data/sub_query.cpp

4.54 SubTask Class Reference

Inheritance diagram for SubTask:



Public Member Functions

- void **execute** () override

Protected Member Functions

- **LEMONDB_QUERY_PTR** (SubQuery)

Friends

- class **SubQuery**

Additional Inherited Members

4.54.1 Detailed Description

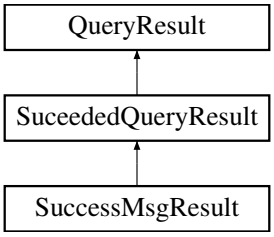
Definition at line 23 of file sub_query.h.

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

- src/query/data/sub_query.h
- src/query/data/sub_query.cpp

4.55 SuccessMsgResult Class Reference

Inheritance diagram for SuccessMsgResult:



Public Member Functions

- **SuccessMsgResult** (const int number)
- **SuccessMsgResult** (std::vector< int > results)
- **SuccessMsgResult** (const char *qname)
- **SuccessMsgResult** (const char *qname, const std::string &msg)
- **SuccessMsgResult** (const char *qname, const char *table, const std::string &msg)
- std::string **toString** () override

Additional Inherited Members

4.55.1 Detailed Description

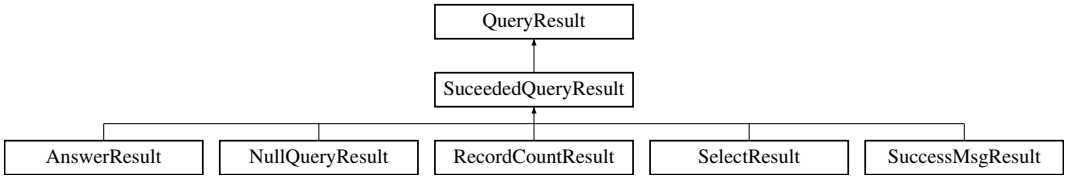
Definition at line 60 of file query_results.h.

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

- src/query_results.h

4.56 SuceededQueryResult Class Reference

Inheritance diagram for SuceededQueryResult:



Public Member Functions

- bool **success** () override

Additional Inherited Members

4.56.1 Detailed Description

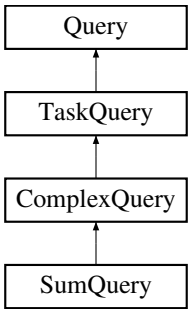
Definition at line 28 of file query_results.h.

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

- src/query_results.h

4.57 SumQuery Class Reference

Inheritance diagram for SumQuery:



Public Member Functions

- **LEMONDB_QUERY_WRITER** (false)
- QueryResult::Ptr **execute** () override
- std::string **toString** () override
- QueryResult::Ptr **combine** (int taskComplete) override

Protected Member Functions

- **LEMONDB_TASK_PTR_DEF** (SumTask)

Friends

- class **SumTask**

Additional Inherited Members

4.57.1 Detailed Description

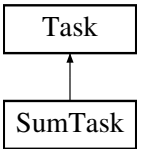
Definition at line 13 of file sum_query.h.

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

- src/query/data/sum_query.h
- src/query/data/sum_query.cpp

4.58 SumTask Class Reference

Inheritance diagram for SumTask:



Public Member Functions

- void **execute** () override

Protected Member Functions

- **LEMONDB_QUERY_PTR** (SumQuery)

Friends

- class **SumQuery**

Additional Inherited Members

4.58.1 Detailed Description

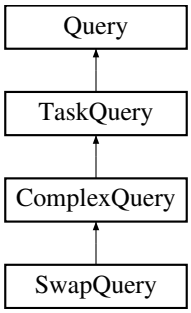
Definition at line 27 of file sum_query.h.

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

- src/query/data/sum_query.h
- src/query/data/sum_query.cpp

4.59 SwapQuery Class Reference

Inheritance diagram for SwapQuery:



Public Member Functions

- **LEMONDB_QUERY_WRITER** (true)
- QueryResult::Ptr **execute** () override
- std::string **toString** () override
- QueryResult::Ptr **combine** (int taskComplete) override

Friends

- class **SwapTask**

Additional Inherited Members

4.59.1 Detailed Description

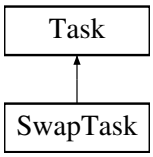
Definition at line 12 of file swap_query.h.

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

- src/query/data/swap_query.h
- src/query/data/swap_query.cpp

4.60 SwapTask Class Reference

Inheritance diagram for SwapTask:



Public Member Functions

- void **execute** () override

Protected Member Functions

- **LEMONDB_QUERY_PTR** (SwapQuery)

Friends

- class **SwapQuery**

Additional Inherited Members

4.60.1 Detailed Description

Definition at line 27 of file swap_query.h.

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

- src/query/data/swap_query.h
- src/query/data/swap_query.cpp

4.61 Table Class Reference

Classes

- class IteratorImpl
- class ObjectImpl

Public Types

- typedef std::string **KeyType**
- typedef std::string **FieldNameType**
- typedef size_t **FieldIndex**
- typedef int **ValueType**
- typedef size_t **SizeType**
- typedef std::unique_ptr< Table > **Ptr**
- typedef ObjectImpl< DataIterator, ValueType > **Object**
- typedef ObjectImpl< ConstDataIterator, const ValueType > **ConstObject**
- typedef IteratorImpl< Object, decltype(data.begin())> **Iterator**
- typedef IteratorImpl< ConstObject, decltype(data.cbegin())> **ConstIterator**

Public Member Functions

- **Table** (const std::string &name)
- template<class FieldIDContainer >
  Table (const std::string &name, const FieldIDContainer &_fields)
- template<class FieldIDContainer >
  void **init** (const FieldIDContainer &fields)
- **Table** (std::string name, const Table &origin)
- void **copy** (const Table &origin)
- void **drop** ()
- bool **isInited** () const
- FieldIndex **getFieldIndex** (const FieldNameType &field) const
- template<class ValueTypeContainer >
  void **insertByIndex** (KeyType key, const ValueTypeContainer &data)
- Object::Ptr operator[ ] (const KeyType &key)
- void eraseUnique (Object::Ptr &&object)
- void erase (const Iterator &it)
- void move (Iterator &it)
- void swapData ()
- bool duplicate (Iterator &it)
- void mergeData ()
- void setName (std::string name)
- const std::string & name () const
- bool empty () const
- size_t size () const
- const std::vector< FieldNameType > & field () const
- size_t clear ()
- Iterator begin ()
- Iterator end ()
- ConstIterator begin () const
- ConstIterator end () const
- void **addQuery** (Query *query)
- void **completeQuery** ()

Static Public Attributes

- static constexpr const ValueType **ValueTypeMax** = INT32_MAX
- static constexpr const ValueType **ValueTypeMin** = INT32_MIN

Friends

- std::ostream & operator<< (std::ostream &os, const Table &table)

4.61.1 Detailed Description

Definition at line 50 of file db_table.h.

4.61.2 Member Function Documentation

4.61.2.1 begin() [1/2]

```
Iterator Table::begin ( )  [inline]
```

Get a begin iterator similar to the standard iterator

Returns

begin iterator

Definition at line 474 of file db_table.h.

4.61.2.2 begin() [2/2]

```
ConstIterator Table::begin ( ) const  [inline]
```

Get a const begin iterator similar to the standard iterator

Returns

const begin iterator

Definition at line 486 of file db_table.h.

4.61.2.3 clear()

```
size_t Table::clear ( )  [inline]
```

Clear all content in the table

Returns

rows affected

Definition at line 463 of file db_table.h.

4.61.2.4 duplicate()

```
bool Table::duplicate (
            Iterator & it )  [inline]
```

Duplicate it and put it into dataNew if {key}_copy exists, nothing happens this function is used only in duplicate query

Definition at line 404 of file db_table.h.

4.61.2.5 empty()

```
bool Table::empty ( ) const  [inline]
```

Return whether the table is empty

Returns

Definition at line 445 of file db_table.h.

4.61.2.6 end() [1/2]

```
Iterator Table::end ( )  [inline]
```

Get a end iterator similar to the standard iterator

Returns

end iterator

Definition at line 480 of file db_table.h.

4.61.2.7 end() [2/2]

```
ConstIterator Table::end ( ) const  [inline]
```

Get a const end iterator similar to the standard iterator

Returns

const end iterator

Definition at line 492 of file db_table.h.

4.61.2.8 erase()

```
void Table::erase (
            const Iterator & it )  [inline]
```

thread safe function Erase the key in the table Caution: this function only erases the key in keyMap, leaves data unchanged no other operation related to keyMap can be applied before swapData is called

Parameters

it	

Definition at line 368 of file db_table.h.

4.61.2.9 eraseUnique()

```
void Table::eraseUnique (
            Object::Ptr && object )  [inline]
```

not thread safe function Remove only one datum only used when delete by key

Parameters

it	

Definition at line 355 of file db_table.h.

4.61.2.10 field()

```
const std::vector<FieldNameType>& Table::field ( ) const  [inline]
```

Return the fields in the table

Returns

Definition at line 457 of file db_table.h.

4.61.2.11 mergeData()

```
void Table::mergeData ( )  [inline]
```

insert dataNew to the end of data then dataNew is cleared for future query this function is used only in duplicate query

Definition at line 421 of file db_table.h.

4.61.2.12 move()

```
void Table::move (
            Iterator & it )  [inline]
```

thread safe function Move datum from data to dataNew Caution: iterator it can't be accessed again after move() is called no other operation related to keyMap can be applied before swapData is called

Parameters

it	

Definition at line 381 of file db_table.h.

4.61.2.13 name()

```
const std::string& Table::name ( ) const  [inline]
```

Get the name of the table

Returns

Definition at line 439 of file db_table.h.

4.61.2.14 operator[]()

```
Object::Ptr Table::operator[] (
            const KeyType & key )  [inline]
```

Access the value according to the key

Parameters

key	

Returns

the Object that KEY = key, or nullptr if key doesn't exist

Definition at line 339 of file db_table.h.

4.61.2.15 setName()

```
void Table::setName (
            std::string name )  [inline]
```

Set the name of the table

Parameters

name	

Definition at line 433 of file db_table.h.

4.61.2.16 size()

```
size_t Table::size ( ) const  [inline]
```

Return the num of data stored in the table

Returns

Definition at line 451 of file db_table.h.

4.61.2.17 swapData()

```
void Table::swapData ( )  [inline]
```

not thread safe function Swap data and newData vector::clear ensures that the capacity of dataNew unchanged so push_back to dataNew is efficient

Definition at line 394 of file db_table.h.

4.61.3 Friends And Related Function Documentation

4.61.3.1 operator<<

```
std::ostream& operator<< (
            std::ostream & os,
            const Table & table )  [friend]
```

Overload the << operator for complete print of the table

Parameters

os	
table	

Returns

the origin ostream

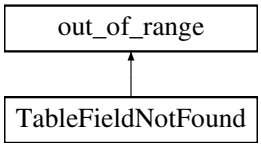
Definition at line 94 of file db_table.cpp.

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

- src/db/db_table.h
- src/db/db_table.cpp

4.62 TableFieldNotFound Struct Reference

Inheritance diagram for TableFieldNotFound:



Public Member Functions

- **TableFieldNotFound** (const std::string &str)

4.62.1 Detailed Description

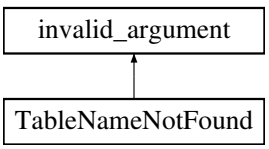
Definition at line 36 of file uexception.h.

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

- src/uexception.h

4.63 TableNameNotFound Struct Reference

Inheritance diagram for TableNameNotFound:



Public Member Functions

- **TableNameNotFound** (const std::string &str)

4.63.1 Detailed Description

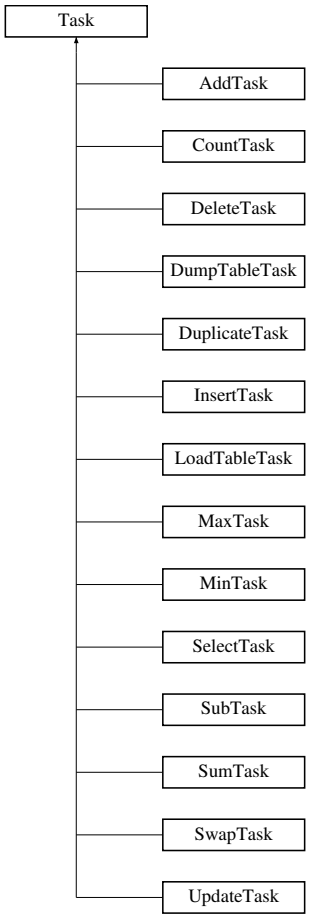
Definition at line 21 of file uexception.h.

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

- src/uexception.h

4.64 Task Class Reference

Inheritance diagram for Task:



Public Types

- typedef std::unique_ptr< Task > **Ptr**

Public Member Functions

- **Task** (Query *query, Table *table=nullptr)
- **Task** (Query *query, Table *table, Table::Iterator begin, Table::Iterator end)
- virtual void **execute** ()
- Table::SizeType **getCounter** () const

Protected Member Functions

- virtual TaskQuery * **getQuery** () const

Protected Attributes

- Query * **query**
- Table * **table** = nullptr
- Table::SizeType counter = 0
- Table::Iterator **begin**
- Table::Iterator **end**
- QueryResult::Ptr **errorResult** = nullptr

Friends

- class **Database**

4.64.1 Detailed Description

Definition at line 12 of file task.h.

4.64.2 Member Data Documentation

4.64.2.1 counter

```
Table::SizeType Task::counter = 0  [protected]
```

Count affected rows in this task

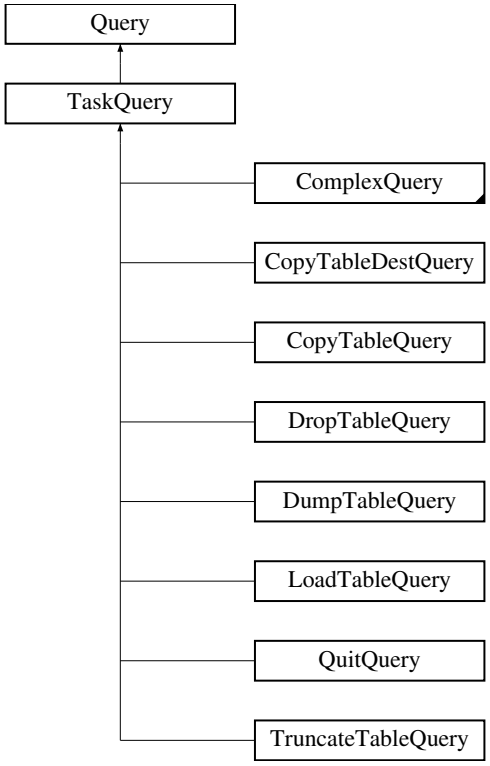
Definition at line 17 of file task.h.

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

- src/query/task.h
- src/query/task.cpp

4.65 TaskQuery Class Reference

Inheritance diagram for TaskQuery:



Public Member Functions

- **TaskQuery** (std::string targetTable)
- Task * **getTask** (size_t index) const
- Task * **getTask** (const std::vector< std::unique_ptr< Task > >::iterator &it) const
- void start ()
- void complete ()
- void complete (QueryResult::Ptr &&result)
- template<class TaskType >
  void addIterationTask (Database &db, Table &table)
- template<class TaskType >
  void addUniqueTask (Database &db, Table *table=nullptr)

Protected Attributes

- size_t tasksSize = 1
- int taskComplete = 0
- std::vector< std::unique_ptr< Task > > tasks
- std::mutex tasksMutex

Additional Inherited Members

4.65.1 Detailed Description

Definition at line 33 of file query.h.

4.65.2 Member Function Documentation

4.65.2.1 addIterationTask()

```
template<class TaskType >
void TaskQuery::addIterationTask (
            Database & db,
            Table & table )  [inline]
```

For iteration query, we can split them in this function

Definition at line 59 of file query.h.

4.65.2.2 addUniqueTask()

```
template<class TaskType >
void TaskQuery::addUniqueTask (
            Database & db,
            Table * table = nullptr )  [inline]
```

For non-iteration query that should be done later

Definition at line 92 of file query.h.

4.65.2.3 complete() [1/2]

```
void TaskQuery::complete ( )
```

Complete a task add the complete query to the result vector here should add a unique id for each query - ok should add a function to print results in correct order

Definition at line 12 of file query.cpp.

4.65.2.4 complete() [2/2]

```
void TaskQuery::complete (
            QueryResult::Ptr && result )
```

Complete a query

Definition at line 26 of file query.cpp.

4.65.2.5 start()

```
void TaskQuery::start ( )
```

Debug function for starting a query

Definition at line 8 of file query.cpp.

4.65.3 Member Data Documentation

4.65.3.1 taskComplete

```
int TaskQuery::taskComplete = 0  [protected]
```

Count the completed tasks, locked by tasksMutex

Definition at line 38 of file query.h.

4.65.3.2 tasks

```
std::vector<std::unique_ptr<Task> > TaskQuery::tasks  [protected]
```

The unique_ptr of tasks are stored here

Definition at line 40 of file query.h.

4.65.3.3 tasksMutex

```
std::mutex TaskQuery::tasksMutex  [protected]
```

protect taskComplete and tasks

Definition at line 42 of file query.h.

4.65.3.4 tasksSize

```
size_t TaskQuery::tasksSize = 1  [protected]
```

The size of tasks, defined to avoid locking

Definition at line 36 of file query.h.

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

- src/query/query.h
- src/query/query.cpp

4.66 TokenizedQueryString Struct Reference

Public Attributes

- std::vector< std::string > **token**
- std::string **rawQeuryString**

4.66.1 Detailed Description

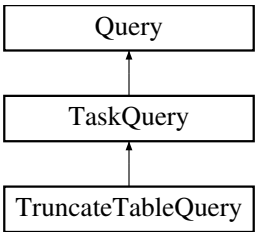
Definition at line 9 of file query_parser.h.

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

- src/query_parser.h

4.67 TruncateTableQuery Class Reference

Inheritance diagram for TruncateTableQuery:



Public Member Functions

- QueryResult::Ptr **execute** () override
- std::string **toString** () override

Additional Inherited Members

4.67.1 Detailed Description

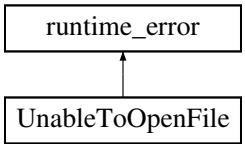
Definition at line 11 of file truncate_query.h.

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

- src/query/management/truncate_query.h
- src/query/management/truncate_query.cpp

4.68 UnableToOpenFile Struct Reference

Inheritance diagram for UnableToOpenFile:



Public Member Functions

- **UnableToOpenFile** (const std::string &file)

4.68.1 Detailed Description

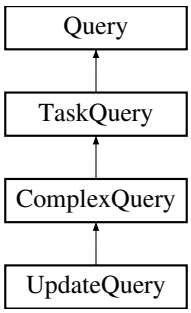
Definition at line 9 of file uexception.h.

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

- src/uexception.h

4.69 UpdateQuery Class Reference

Inheritance diagram for UpdateQuery:



Public Member Functions

- **LEMONDB_QUERY_WRITER** (true)
- QueryResult::Ptr **execute** () override
- std::string **toString** () override
- QueryResult::Ptr **combine** (int taskComplete) override

Friends

- class **UpdateTask**

Additional Inherited Members

4.69.1 Detailed Description

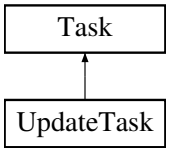
Definition at line 7 of file update_query.h.

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

- src/query/data/update_query.h
- src/query/data/update_query.cpp

4.70 UpdateTask Class Reference

Inheritance diagram for UpdateTask:



Public Member Functions

- void **execute** () override

Protected Member Functions

- **LEMONDB_QUERY_PTR** (UpdateQuery)

Friends

- class **ComplexQuery**

Additional Inherited Members

4.70.1 Detailed Description

Definition at line 24 of file update_query.h.

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

- src/query/data/update_query.h
- src/query/data/update_query.cpp

Index

addIterationTask
    TaskQuery, 65
AddQuery, 9
addQuery
    Database, 20
AddTask, 10
addTask
    Database, 20
addUniqueTask
    TaskQuery, 65
AnswerResult, 11

BasicQueryBuilder, 11
begin
    Table, 56

clear
    Table, 56
complete
    TaskQuery, 65
completeQuery
    Database, 20
ComplexQuery, 12
    condition, 15
    evalCondition, 13
    getCondition, 13
    getOperands, 13
    initCondition, 14
    operands, 15
    testKeyCondition, 14
ComplexQueryBuilder, 15
condition
    ComplexQuery, 15
ConflictingKey, 16
CopyTableDestQuery, 16
CopyTableQuery, 17
CountQuery, 18
CountTask, 19
counter
    Task, 63

Database, 19
    addQuery, 20
    addTask, 20
    completeQuery, 20
    ensureTable, 21
DeleteQuery, 21
DeleteTask, 22
DropTableQuery, 23
DumpTableQuery, 23
DumpTableTask, 24
duplicate
    Table, 56
DuplicateQuery, 25
DuplicateTask, 26
DuplicatedTableName, 25

empty
    Table, 57
end
    Table, 57
ensureTable
    Database, 21
erase
    Table, 57
eraseUnique
    Table, 58
ErrorMsgResult, 27
evalCondition
    ComplexQuery, 13

FailedQueryBuilder, 27
FailedQueryResult, 28
field
    Table, 58

getCondition
    ComplexQuery, 13
getId
    Query, 42
getOperands
    ComplexQuery, 13

IllFormedQuery, 29
IllFormedQueryCondition, 29
initCondition
    ComplexQuery, 14
initId
    Query, 42
InsertQuery, 30
InsertTask, 30

ListTableQuery, 32
LoadFromStreamException, 32
LoadTableQuery, 33
LoadTableTask, 34

MaxQuery, 34
MaxTask, 35
mergeData
    Table, 58

MinQuery, 36
MinTask, 37
move
    Table, 58
MultipleKey, 38

name
    Table, 59
NopQuery, 38
NullQueryResult, 39

operands
    ComplexQuery, 15
operator<<
    Table, 60
operator[]
    Table, 59

PrintTableQuery, 40

Query, 41
    getId, 42
    initId, 42
QueryBuilder, 42
QueryBuilderMatchFailed, 43
QueryCondition, 44
QueryParser, 44
QueryResult, 44
QuitQuery, 45

RecordCountResult, 46

SelectQuery, 46
SelectResult, 47
SelectTask, 48
setName
    Table, 59
size
    Table, 60
start
    TaskQuery, 65
SubQuery, 48
SubTask, 49
SuccessMsgResult, 50
SuceededQueryResult, 51
SumQuery, 51
SumTask, 52
swapData
    Table, 60
SwapQuery, 53
SwapTask, 54

Table, 54
    begin, 56
    clear, 56
    duplicate, 56
    empty, 57
    end, 57
    erase, 57
    eraseUnique, 58
    field, 58
    mergeData, 58
    move, 58
    name, 59
    operator<<, 60
    operator[], 59
    setName, 59
    size, 60
    swapData, 60
Table::IteratorImpl< ObjType, DatumIterator >, 31
Table::ObjectImpl< Iterator, VType >, 39
TableFieldNotFound, 61
TableNameNotFound, 61
Task, 62
    counter, 63
taskComplete
    TaskQuery, 66
TaskQuery, 64
    addIterationTask, 65
    addUniqueTask, 65
    complete, 65
    start, 65
    taskComplete, 66
    tasks, 66
    tasksMutex, 66
    tasksSize, 66
tasks
    TaskQuery, 66
tasksMutex
    TaskQuery, 66
tasksSize
    TaskQuery, 66
testKeyCondition
    ComplexQuery, 14
TokenizedQueryString, 67
TruncateTableQuery, 67

UnableToOpenFile, 68
UpdateQuery, 68
UpdateTask, 69