

*Fork me on GitHub*

Api

SELECT

DELETE

INSERT

UPDATE

Functions

Misc

SELECT

The following structure defines the SQL query syntax in LiteDB. Keywords and function names are case-insensitive.

```
[ EXPLAIN ]
  SELECT    <selectExpr0> [, <selectExprN>]
[ INTO      <collection>|<$file(param)> [ : <autoIdType> ] ]
[ FROM      <collection>|<$systemCollection> ]
[ INCLUDE    <pathExpr0> [, <pathExprN> ]
[ WHERE      <filterExpr> ]
[ GROUP BY   <groupByExpr>} ]
[ HAVING     <filterExpr> ]
[ ORDER BY   <orderByExpr> [ ASC | DESC ] ]
[ LIMIT      <number> ]
[ OFFSET     <number> ]
[ FOR UPDATE ]
```

Explain

If the keyword `EXPLAIN` is present before a query, the result is a document that explains how the engine plans to run the query.

Select

The `SELECT` clause defines the projections that are applied to the results. A select expression can be:

- A literal of any BSON type that LiteDB supports;

- A valid JSON path;
- A function over literals or JSON paths.

The `GROUP BY` clause restricts the possible values in this clause. For more info, check the `GROUP BY` documentation below.

Into

If this clause is present, the result of the query is inserted into `collection` and the query returns the number of documents inserted.

If the result does not contain a `$_id` field, `autoIdType` is used to generate one of the specified type (`GUID` , `INT` , `LONG` or `OBJECTID`). If no `autoIdType` is present, the default is `OBJECTID` .

If `collection` is the system collection `$file` , the result will be written to a file. For more information, check the System Collections documentation.

From

This clause is used to indicate the source collection for the query.

If `collection` starts with `$` , it is a system collection. For more information, check the System Collections documentation.

Include

If this clause is presents, references are resolved by the query engine before returning the results.

Every path expression must be a valid JSON path that points to an embedded document or array of embedded documents with the following structure:

```
{
    "$id" : 1,
    "$ref" : "Customers"
}
```

The field `$ref` indicates which collection is being referenced and the field `$id` corresponds to the `$_id` field in the document being referenced.

A filter expression is a series of expressions joined by the logical operators `AND` or `OR` . Every expression must resolve to a boolean value. For more information, check the [Expressions documentation](#).

Group By

If this clause is present, the results are grouped by an expression and the query returns a document for each group.

A group-by expression can be any valid expression. For more information, check the Expressions documentation.

Please note that only one grouping expression is allowed. However, grouping by multiple fields can be achieved by using a document or array containing the desired fields as the grouping expression.

If this clause is present, only the special parameter `@key` (which returns the value in the grouping expression) or aggregate functions can be used in the `SELECT` clause.

Having

If this clause is present, the groups are filtered by a filter expression (similarly to how a `WHERE` clause filters individual documents).

A filter expression is a series of expressions joined by the logical operators `AND` or `OR`. Every expression must resolve to a boolean value. For more information, check the Expressions documentation.

In the `HAVING` clause, only the special parameter `@key` (which returns the value in the grouping expression) or aggregate functions can be used.

Order By

If this clause is present, the resulting documents are ordered by the provided sort expression.

A sort expression can be any valid expression. For more information, check the Expressions documentation.

This clause cannot be used with a `GROUP BY` clause.

Limit

Limits the amount of documents return to a maximum of `number`.

Offset

Returns only the documents starting from the position `number`. Please note that the offset value is zero-based.

For Update

If this clause is present, the collections used in the query are opened with full write lock.

Made with ♥ by LiteDB team - @mbdavid - MIT License