
mysqlparse Documentation

Release 0.1.6

Julius Šėporaitis

Sep 27, 2017

Contents

1	Contents
----------	-----------------

3

The purpose of `mysqlparse` library is to provide structural access to MySQL queries. It looks better with an example:

```
>>> import mysqlparse
>>> sql = mysqlparse.parse("""
...     ALTER TABLE `django_user` ADD COLUMN `notes` LONGTEXT NOT NULL
... """)
>>> print(sql.statements[0].statement_type)
ALTER
>>> print(sql.statements[0].table_name)
`django_user`
>>> print(sql.statements[0].ignore)
False
>>> print(sql.statements[0].alter_specification[0].alter_action)
ADD COLUMN
>>> print(sql.statements[0].alter_specification[0].column_name)
`notes`
>>> print(sql.statements[0].alter_specification[0].data_type)
LONGTEXT
>>> print(sql.statements[0].alter_specification[0].null)
False
>>> print(sql.statements[0].alter_specification[0].column_position)
LAST
```


Requirements and Installation

Requirements

Tested with:

- Python: 2.7, 3.3, 3.4, 3.5.
- pyparsing: 2.2.0+
- six: 1.10.0+

Installation

Install with pip:

```
$ pip install mysqlparse
```

Or add it to your project's `requirements.txt`.

Supported Statements

This is how `mysqlparse` coverage compares to `MySQL ALTER TABLE` statement syntax:

```
ALTER TABLE tbl_name
    [alter_specification [, alter_specification] ...]

alter_specification:
    ADD [COLUMN] col_name column_definition
        [FIRST | AFTER col_name]
    | ADD {INDEX | KEY} [index_name]
```

```

    [index_type] (index_col_name, ...) [index_option] ...
| CHANGE [COLUMN] old_col_name new_col_name column_definition
    [FIRST | AFTER col_name]
| MODIFY [COLUMN] col_name column_definition
    [FIRST | AFTER col_name]
| DROP [COLUMN] col_name
| DROP PRIMARY KEY
| DROP {INDEX | KEY} index_name
| DROP FOREIGN KEY fk_symbol

index_type:
    USING {BTREE | HASH}

index_option:
    KEY_BLOCK_SIZE [=] value
| index_type
| WITH PARSER parser_name
| COMMENT 'string'

column_definition:
    data_type [NOT NULL | NULL] [DEFAULT default_value]
    [AUTO_INCREMENT] [UNIQUE [KEY] | [PRIMARY] KEY]
    [COMMENT 'string']

data_type:
    BIT[(length)]
| TINYINT[(length)] [UNSIGNED] [ZEROFILL]
| SMALLINT[(length)] [UNSIGNED] [ZEROFILL]
| MEDIUMINT[(length)] [UNSIGNED] [ZEROFILL]
| INT[(length)] [UNSIGNED] [ZEROFILL]
| INTEGER[(length)] [UNSIGNED] [ZEROFILL]
| BIGINT[(length)] [UNSIGNED] [ZEROFILL]
| REAL[(length,decimals)] [UNSIGNED] [ZEROFILL]
| DOUBLE[(length,decimals)] [UNSIGNED] [ZEROFILL]
| FLOAT[(length,decimals)] [UNSIGNED] [ZEROFILL]
| DECIMAL[(length[,decimals])] [UNSIGNED] [ZEROFILL]
| NUMERIC[(length[,decimals])] [UNSIGNED] [ZEROFILL]
| DATE
| TIME[(fsp)]
| TIMESTAMP[(fsp)]
| DATETIME[(fsp)]
| YEAR
| CHAR[(length)] [BINARY]
    [CHARACTER SET charset_name] [COLLATE collation_name]
| VARCHAR(length) [BINARY]
    [CHARACTER SET charset_name] [COLLATE collation_name]
| BINARY[(length)]
| VARBINARY(length)
| TINYBLOB
| BLOB
| MEDIUMBLOB
| LONGBLOB
| TINYTEXT [BINARY]
    [CHARACTER SET charset_name] [COLLATE collation_name]
| TEXT [BINARY]
    [CHARACTER SET charset_name] [COLLATE collation_name]
| MEDIUMTEXT [BINARY]
    [CHARACTER SET charset_name] [COLLATE collation_name]

```

```

| LONGTEXT [BINARY]
  [CHARACTER SET charset_name] [COLLATE collation_name]
| ENUM(value1,value2,value3,...)
  [CHARACTER SET charset_name] [COLLATE collation_name]
| SET(value1,value2,value3,...)
  [CHARACTER SET charset_name] [COLLATE collation_name]

```

This is how mysqlparse coverage compares to **MySQL CREATE TABLE** statement syntax:

```

CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
  (create_definition,...)
  [table_options]

create_definition:
  col_name column_definition

column_definition:
  data_type [NOT NULL | NULL] [DEFAULT default_value]
    [AUTO_INCREMENT] [UNIQUE [KEY] | [PRIMARY] KEY]
    [COMMENT 'string']

data_type:
  BIT[(length)]
| TINYINT[(length)] [UNSIGNED] [ZEROFILL]
| SMALLINT[(length)] [UNSIGNED] [ZEROFILL]
| MEDIUMINT[(length)] [UNSIGNED] [ZEROFILL]
| INT[(length)] [UNSIGNED] [ZEROFILL]
| INTEGER[(length)] [UNSIGNED] [ZEROFILL]
| BIGINT[(length)] [UNSIGNED] [ZEROFILL]
| REAL[(length,decimals)] [UNSIGNED] [ZEROFILL]
| DOUBLE[(length,decimals)] [UNSIGNED] [ZEROFILL]
| FLOAT[(length,decimals)] [UNSIGNED] [ZEROFILL]
| DECIMAL[(length[,decimals])] [UNSIGNED] [ZEROFILL]
| NUMERIC[(length[,decimals])] [UNSIGNED] [ZEROFILL]
| DATE
| TIME[(fsp)]
| TIMESTAMP[(fsp)]
| DATETIME[(fsp)]
| YEAR
| CHAR[(length)] [BINARY]
  [CHARACTER SET charset_name] [COLLATE collation_name]
| VARCHAR(length) [BINARY]
  [CHARACTER SET charset_name] [COLLATE collation_name]
| BINARY[(length)]
| VARBINARY(length)
| TINYBLOB
| BLOB
| MEDIUMBLOB
| LONGBLOB
| TINYTEXT [BINARY]
  [CHARACTER SET charset_name] [COLLATE collation_name]
| TEXT [BINARY]
  [CHARACTER SET charset_name] [COLLATE collation_name]
| MEDIUMTEXT [BINARY]
  [CHARACTER SET charset_name] [COLLATE collation_name]
| LONGTEXT [BINARY]
  [CHARACTER SET charset_name] [COLLATE collation_name]
| ENUM(value1,value2,value3,...)

```

```

    [CHARACTER SET charset_name] [COLLATE collation_name]
  | SET(value1,value2,value3,...)
    [CHARACTER SET charset_name] [COLLATE collation_name]

table_option:
  ENGINE [=] engine_name
  | AUTO_INCREMENT [=] value
  | AVG_ROW_LENGTH [=] value
  | [DEFAULT] CHARACTER SET [=] charset_name
  | CHECKSUM [=] {0 | 1}
  | [DEFAULT] COLLATE [=] collation_name
  | COMMENT [=] 'string'
  | COMPRESSION [=] {'ZLIB'|'LZ4'|'NONE'}
  | CONNECTION [=] 'connect_string'
  | DATA DIRECTORY [=] 'absolute path to directory'
  | DELAY_KEY_WRITE [=] {0 | 1}
  | ENCRYPTION [=] {'Y' | 'N'}
  | INDEX DIRECTORY [=] 'absolute path to directory'
  | INSERT_METHOD [=] { NO | FIRST | LAST }
  | KEY_BLOCK_SIZE [=] value
  | MAX_ROWS [=] value
  | MIN_ROWS [=] value
  | PACK_KEYS [=] {0 | 1 | DEFAULT}
  | PASSWORD [=] 'string'
  | ROW_FORMAT [=] {DEFAULT|DYNAMIC|FIXED|COMPRESSED|REDUNDANT|COMPACT}
  | STATS_AUTO_RECALC [=] {DEFAULT|0|1}
  | STATS_PERSISTENT [=] {DEFAULT|0|1}
  | STATS_SAMPLE_PAGES [=] value
  | TABLESPACE tablespace_name [STORAGE {DISK|MEMORY|DEFAULT}]
  | UNION [=] (tbl_name[,tbl_name]...)

```

Statements Structure

`mysqlparse.parse(file_or_string)`

Takes a file like object or string and returns `pyarsing.ParseResults` representing the SQL file structure.

Assuming sql file is parsed like this:

```

>>> with open('001_migrate.sql') as sql_file:
>>>     sql = mysqlparse.parse(sql_file)

```

The following properties are accessible

- `sql.statements[]` : list all individual sql statements, separated by ; are accessible through this list.
- `.statement_type` : str one of:
 - * ALTER
 - * CREATE
- `.create_type` : str, currently only TABLE.
- `.temporary` : boolean, True if CREATE TEMPORARY TABLE.
- `.overwrite` : boolean, False when IF NOT EXISTS is present.

- `.database_name` : str, None or database. name if the table identifier was with a dot (e.g. `db_name.tbl_name`).
- `.table_name` : str, table name of ALTER TABLE statement.
- `.ignore` : boolean, True if it is ALTER IGNORE TABLE statement (support for it is removed as of MySQL 5.7.4).
- `.table_options[]` : list list with key and value pairs for table options:
 - * `.key` key
 - * `.value` value
- `.alter_specification[]` : list list of individual column alterations.
 - * `.alter_action` : str one of:
 - ADD COLUMN
 - ADD INDEX
 - MODIFY COLUMN
 - CHANGE COLUMN
 - DROP COLUMN
 - DROP PRIMARY KEY
 - DROP INDEX
 - DROP KEY
 - DROP FOREIGN KEY.
 - * `.column_name` : str
 - * `.new_column_name` : str name of the new column name in MODIFY COLUMN statements.
 - * `.null` : boolean|str - True if the column is null, False - if not null and implicit if unspecified.
 - * `.default` : str - default value of the column.
 - * `.auto_increment` : boolean - True if the column is auto increment.
 - * `.index_type` : str - `unique_key` if column is unique key, `primary_key` if column is primary key, `BTREE` if it is `btree`, `HASH` if it is has.
 - * `.key_block_size` : str key block size of index.
 - * `.parser_name` : str name of the parser.
 - * `.comment` : str - comment string.
 - * `.column_position` : str one of: `FIRST`, another column name or (default) `LAST`.
 - * `.data_type` : str one of:
 - BIT
 - TINYINT
 - SMALLINT
 - MEDIUMINT
 - INT

- INTEGER
 - BIGINT
 - REAL
 - DOUBLE
 - FLOAT
 - DECIMAL
 - NUMERIC
 - DATE
 - TIME
 - TIMESTAMP
 - DATETIME
 - YEAR
 - CHAR
 - VARCHAR
 - BINARY
 - VARBINARY
 - TINYBLOB
 - BLOB
 - MEDIUMBLOB
 - LONGBLOB
 - TINYTEXT
 - TEXT
 - MEDIUMTEXT
 - LONGTEXT
 - ENUM
 - SET
- * `.length` : str - column length (as in `INT(length)`).
 - * `.decimals` : str - number of decimal places of a decimal type.
 - * `.unsigned` : boolean - True if column is of `UNSIGNED` type.
 - * `.zerofill` : boolean - True if column is of `ZEROFILL` type.
 - * `.binary` : boolean - True if column is of `BINARY` type.
 - * `.character_set` : str - character set of the column.
 - * `.collate` : str - column collation name.