

---

# **agate-sql Documentation**

*Release 0.3.0 (alpha)*

**Christopher Groskopf**

November 05, 2015



<b>1</b>	<b>Install</b>	<b>3</b>
<b>2</b>	<b>Usage</b>	<b>5</b>
<b>3</b>	<b>API</b>	<b>7</b>
3.1	Authors . . . . .	7
3.2	Changelog . . . . .	8
3.3	License . . . . .	8
3.4	Indices and tables . . . . .	8



agate-sql adds SQL read/write support to agate.

Important links:

- agate <http://agate.rtfid.org>
- Documentation: <http://agate-sql.rtfid.org>
- Repository: <https://github.com/onyxfish/agate-sql>
- Issues: <https://github.com/onyxfish/agate-sql/issues>



---

## Install

---

To install:

```
pip install agatesql
```

For details on development or supported platforms see the [agate documentation](#).

**Warning:** You'll need to have the correct [sqlalchemy drivers](#) installed for whatever database you plan to access. For instance, in order to read/write tables in a Postgres database, you'll also need to `pip install psycopg2`.





---

## Usage

---

`agate-sql` uses a monkey patching pattern to add SQL support to all `agate.Table` instances.

```
import agate
import agatesql

agatesql.patch()
```

Calling `patch()` attaches all the methods of `TableSQL` to `agate.Table`. For example, to import a table named `doctors` from a local postgresql database named `hospitals` you will use `TableSQL.from_sql()`:

```
new_table = agate.Table.from_sql('postgresql:///hospitals', 'doctors')
```

To save this table back to the database:

```
new_table.to_sql('postgresql:///hospitals', 'doctors')
```

The first argument to either function can be any valid `sqlalchemy` connection string. The second argument must be a database name. (Arbitrary SQL queries are not supported.)

That's all there is to it.



---

`agate.sql.patch()`

Patch the features of this library onto agate's core `Table` and `TableSet`.

**class** `agate.sql.table.TableSQL`

**classmethod** `from_sql` (*connection\_or\_string*, *table\_name*)

Create a new `agate.Table` from a given SQL table. Types will be inferred from the database schema.

Monkey patched as class method `Table.from_sql()`.

**Parameters**

- **connection\_or\_string** – An existing sqlalchemy connection or a connection string.
- **table\_name** – The name of a table in the referenced database.

**to\_sql** (*connection\_or\_string*, *table\_name*, *overwrite=False*)

Write this table to the given SQL database.

Monkey patched as instance method `Table.to_sql()`.

**Parameters**

- **connection\_or\_string** – An existing sqlalchemy connection or a connection string.
- **table\_name** – The name of the SQL table to create.
- **overwrite** – If `True`, any existing table with the same name will be dropped and recreated.

## 3.1 Authors

The following individuals have contributed code to `agate-sql`:

- Christopher Groskopf
- Adrian Klaver

## 3.2 Changelog

### 3.2.1 0.3.0 - November 5, 2015

- Add `overwrite` flag to `TableSQL.to_sql()`.
- Removed Python 2.6 support.
- Updated agate dependency to version 1.1.0.
- Additional SQL types are now supported. (#4, #10)

### 3.2.2 0.2.0 - October 22, 2015

- Add explicit patch function.

### 3.2.3 0.1.0 - September 22, 2015

- Initial version.

## 3.3 License

The MIT License

Copyright (c) 2015 Christopher Groskopf and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 3.4 Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

## F

`from_sql()` (`agatesql.table.TableSQL` class method), 7

## P

`patch()` (in module `agatesql`), 7

## T

`TableSQL` (class in `agatesql.table`), 7

`to_sql()` (`agatesql.table.TableSQL` method), 7