

---

# **BeeSQL2 Documentation**

*Release 0.1*

**Kasun Herath**

December 26, 2016



<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>UserGuide</b>	<b>3</b>
2.1	Quick Start . . . . .	3
<b>3</b>	<b>API</b>	<b>5</b>
3.1	beesql . . . . .	5
3.2	MySQL . . . . .	5
3.3	SQLite . . . . .	5
	<b>Python Module Index</b>	<b>7</b>



---

# Introduction

---

BeeSQL2 is a SQL abstraction library targetting python that helps,

- Minimize repetitive steps in Python DB-API.
- Use python methods for SQL generation.
- Map SQL to Python datastructures.

BeeSQL2 is not an ORM.

Currently BeeSQL supports following databases.

- MySQL



## 2.1 Quick Start

**Note:** Current supported engines are `mysql`.

### 2.1.1 Common operations

**Importing beesql:**

```
import beesql
```

**Creating a database connection:**

```
from beesql import DB

db = DB(database_type='mysql', db_name="database_name", username="username" password='password',
        host="database_host", port="database_port")

db = DB('mysql', 'db_name').auth('username', 'password')
```

The `database_type` should be one of supported engines.

**\*\* Creating a statement \*\*:**

```
statement = db.query('table_name').select()
statement = db.query('table_name').update(name="new name")
statement = db.query('table_name').delete()
```

**\*\* Select statement \*\*:**

- `statement = db.query().on('table_name').select() => "SELECT * FROM table_name"`
- `statement = db.query().on('table_name').select('id', 'age') => "SELECT id, age FROM table_name"`
- `statement = db.query('table_name').select('id', 'age').select('location') => "SELECT id, age, location FROM table_name"`

**\*\* Update statement \*\*:**

- `statement = db.query('table_name').update(age=23) => "UPDATE table_name SET age=23"`

**\*\* Delete statement \*\*:**

- `statement = db.query('table_name').delete() => '' DELETE FROM table_name ''`

**\*\* Where condition \*\*::**

- `db.query('table_name').select().where(age=20, code='100') => '' SELECT * FROM table_name WHERE age = 20 AND code = 100`
- `db.query('table_name').select().where(age=20)._and('code').eq(100) => '' SELECT * FROM table_name WHERE age = 20 AND code = 100`
- `db.query('table_name').select().where('age').lt(100)._or('code').gte(10) => '' SELECT * FROM table_name WHERE age < 20 OR code >= 100`



### 3.1 beesql

BeeSQL operations are executed through a database connection. Create a database connection using `connection()`.

### 3.2 MySQL

MySQL connection is used to operate on a MySQL database.

### 3.3 SQLite

SQLite connection is used to operate on a SQLite database.



**b**

beesql, 5  
beesql.backends.mysql, 5  
beesql.backends.sqlite, 5



## B

beesql (module), 5

beesql.backends.mysql (module), 5

beesql.backends.sqlite (module), 5