Python-MySQL-Connector-Wrapper-Class

Wrapper Class to help you build mysql queries for the offical python mysql connector.

How to use

===

Initise Class

```
import Sql
sql = Sql()
```

insert Method

```
sql.insert("my_table",{'username':'John','email':'example@gmail.com'})
print sql.insert_id
```

Select Method

```
sql.select("your_table")
data = sql.fetch()
print data
```

Select Method with where clause

```
sql.select("my_table","*",'where id = 1')
data = sql.fetch()
print data

sql.select("my_table",'*','where id = %(id)s', {'id':10})
data = sql.fetch
print data

sql.select("my_table",'*','where id = %(id)s and visible =
%(visible)s', {'id':10,'visible':1})
data = sql.fetch()
print data
```

Select Method parsing start, limit, order, direction

```
sql.select("user_table",'*','', {}, 0 , 20,'timestamp','des
c')
data = sql.fetch()
print data

sql.select("my_table",'*','where id = %(id)s and visible =
%(visible)s', {'id':10,'visible':1}, 0, 20, 'timestamp', 'd
esc')
data = sql.fetch()
print data
```

Specify fields in table

```
sql.select("my_table",' id')
data = sql.fetch()
print data

sql.select("my_table",'email, id')
data = sql.fetch()
print data

sql.select("my_table",'email, id','where id = %(id)s or vis
ible = %(visible)s', {'id':10, 'visible':1}, 0 , 20)
data = sql.fetch()
print data

sql.select("user_table",'count(*) as total')
data = sql.fetch()
print data
```

Select Where Method

```
sql.where("my_table", {"id": 1})
data = sql.fetch()
print data

sql.where("my_table", {"id": 10,'visible':1})
data = sql.fetch()
```

```
print data
```

Select Where Method parsing start, limit, order, direction

```
sql.where("my_table", {"id": 10,'visible':1}, 0 , 20,'times
tamp','desc')
data = sql.fetch()
print data
```

Select Where Method using more comparison

```
search_array = {
    'visible':1,
    'sex':'female'
}
sql.where("my_table", search_array , 0 , 20,'timestamp','de
sc')
data = sql.fetch()
print data

search_array = {
    'status !=':1,
    'DATE_FORMAT(dispatch_time,"%Y-%m-%d %H:%i:%s") < ':str
ftime("%Y-%m-%d %H:%M:%S", gmtime())
}
sql.where("my_table", search_array , 0 , 20,'timestamp','de
sc')
data = sql.fetch()
print data

search_array = {
    'age >=':20,
    'sex':'male'
}
sql.where("my_table", search_array , 0 , 20,'timestamp','de
sc')
data = sql.fetch()
print data
```

Select Where Method using parsing field to be selected

```
search_array = {
    'status !=':1,
}
self.sql.field = "id, email"
sql.where("my_table", search_array , 0 , 20,'timestamp','de
sc')
data = sql.fetch()
print data

search_array = {
    'paid':1,
    'title':'Alex'
}
sql._field(['id','status'])
sql.where("my_table", search_array , 0 , 20,'timestamp','de
sc')
data = sql.fetch()
print data
```

Update Method

```
status = sql.update("my_table",{'status':'0'})
status = sql.update("my_table",{'username':'John','sex':'fe
male'},'where id = %(id)s and sex = %(sex)s ',{'id':10,'sex
':'male'})
status = sql.update("my_table",{'username':'John','sex':'fe
male'},'where id = %(id)s',{'id':10})
```

Update Method using only dictionary conditions

```
conditions = {'id':10,'sex':'male'}
status = sql.update("my_table",{'username':'John','sex':'fe
male'},conditions)
conditions = {
    'age >='210,
    'sex':'male'
}
status = sql.update("my_table",{'username':'John','sex':'fe
male'},conditions)
```

Delete Method

```
status = sql.delete("my_table",'where id = %(id)s ',{'id':1
0})
status = sql.delete("my_table",'where id = %(id)s and sex =
 %(sex)s ',{'id':10,'sex':'male'})
```

Update Method using only dictionary conditions

```
conditions = {'id':10,'sex':'male'}
status = sql.delete("my_table",conditions)

conditions = {
    'age >='210,
    'sex':'male'
}
status = sql.update("my_table",conditions)
```

A lot can still be done, submit a pull request and add your contribution