

Yii2 Select Query : Model

This tutorial will help you to get the list of function to retrieve or select the data from the database using model of yii2.0 framework. I updated most of the functions it will helpful for yii2.0 select query method.

1. find()
2. select()
3. all()
4. one()
5. where()
6. orderBy()
7. count()
8. asArray()
9. indexBy()
10. limit()
11. offset()
12. Limit With Pagination()
13. LIKE Condition
14. In Condition
15. between
16. groupBy()
17. having()
18. addParams()
19. Multiple Conditions
20. findBySql()

Examples

find()

Find a single record using primary key. find() method is used for this. If record is available it will return object else we will get false or null;

1	\$model = User::find(1);
2	if(\$model) {
3	echo \$model->username;
4	echo \$model->status;
5	}

select()

select() method is used to select the display columns from the tables of yii2;

1	\$model = User::find()->select('column1, column2')->all();
---	--

all()

Get all the records from the database table using all() method of yii2.0 model.

1	\$model = User::find()->all();
---	--------------------------------

one()

Retrieve the single record from the database based on condition.

1	\$model = User::find()->one();
---	--------------------------------

where()

Sample 1:

1	\$userid=1;
2	\$model = User::find()
3	->where('userid > :userid', [':userid' => \$userid])
4	->one();

Sample 2:

1	\$model = User::find()
2	->where(['reg_date' => \$date, 'status' => 1])
3	->one();

Sample 3:

1	\$model = User::find()
2	->where("reg_date > '2014-01-01' and status=1")
3	->all ();

Sample 4:

1	\$model = User::find()
2	->where('userid > :userid', [':userid' => \$userid])
3	->orWhere('primary_user = :primary_user', [':primary_user' => \$primary_user])
4	->andWhere('status = :status', [':status' => \$status])
5	->all();

OUTPUT Query

```
SELECT * FROM `tbl_user` WHERE ((userid > 1) OR (primary_user = 1)) AND (status = 1)
```

orderBy()

Sample 1:

1	\$model = User::find()
2	->where(['status' => 1])
3	->orderBy('userid')
5	->all();

Sample 2:

1	\$model = User::find()
2	->where(['status' => 0])
3	->orderBy('userid')
5	->one();

Sample 3:

1	\$model = User::find()
2	->orderBy([
3	'usertype'=>SORT_ASC,
5	'username' => SORT_DESC,
6])
7	->limit(10)
8	->all();

OUTPUT Query

```
SELECT * FROM `tbl_user` ORDER BY `usertype`, `username` DESC LIMIT 10
```

count()

1	\$model = User::find()
2	->where(['status' => 0])
3	->orderBy('userid')
4	->count();

asArray()

Sample 1:

1	\$model = User::find()
2	-> asArray()
3	->all()

Sample 2:

1	\$model = User::find()
2	-> asArray()
3	->one()

indexBy()

1	\$model = User::find()
2	-> indexBy('id')
3	->one()

limit()

Sample 1:

1	\$model = User::find()
2	-> limit(10)
3	->all()

Sample 2:

1	\$model = User::find()
2	->where('userid > 1 and isactive=1')
3	->limit(2)
4	->all();

offset()

1	\$model = User::find()
2	->limit(5)
3	->offset(10)
4	->all();

OUTPUT Query

```
SELECT * FROM `tbl_user` LIMIT 5 OFFSET 10
```

Limit With Pagination

Here We limited the record using pagination class and we set the default page size. See the below code It contains total records count, default page size, limit and offset.

1	\$query = Country::find();
2	\$pagination = new Pagination([
3	'defaultPageSize' => 5,
4	'totalCount' => \$query->count(),
5]);
6	\$countries = \$query->orderBy('name')
7	->offset(\$pagination->offset)
8	->limit(\$pagination->limit)
9	->all();

LIKE Condition

Sample 1:

1	\$model = User::find()
2	->where(['LIKE', 'username', 'admin'])
3	->all()
4	//OR
5	\$model = User::find()
6	->where('username LIKE :query')
7	->addParams([':query'=>'%admin%'])
8	->all();

Sample 2:

1	\$model = User::find()
2	->where(['NOT LIKE', 'username', 'admin'])
3	->all()

OUTPUT Query

```
SELECT * FROM `tbl_user` WHERE `username` LIKE '%admin%'
SELECT * FROM `tbl_user` WHERE `username` NOT LIKE '%admin%'
```

In Condition

Sample 1:

1	\$model = User::find()
2	->where([
3	'userid' => [1001,1002,1003,1004,1005],
4])
5	->all();

Sample 2:

1	\$model = User::find()
2	->where(['IN', 'userid', [1001,1002,1003,1004,1005]])
3	->all();

OUTPUT Query

```
SELECT * FROM `tbl_user` WHERE `userid` IN (1001, 1002, 1003, 1004, 1005)
```

Sample 3:

1	\$model = User::find()
2	->where(['NOT IN', 'userid', [1001,1002,1003,1004,1005]])
3	->all();

OUTPUT Query

```
SELECT * FROM `tbl_user` WHERE `userid` NOT IN (1001, 1002, 1003, 1004, 1005)
```

between()

Sample 1:

1	\$model = User::find()
2	->select('username')
3	->asArray()
4	->where('userid between 1 and 5')
5	->all();

OUTPUT Query

```
SELECT `username` FROM `tbl_user` WHERE userid between 1 and 5
```

groupBy()

1	\$model = User::find()
2	->groupBy('usertype')
3	->all();

OUTPUT Query

```
SELECT * FROM `tbl_user` GROUP BY `usertype`
```

having()

1	\$states=1;
2	\$model = User::find()
3	->groupBy('usertype')
4	->having('states >:states')
5	->addParams([':states'=>\$states])
6	->all();

OUTPUT Query

```
SELECT * FROM `tbl_user` GROUP BY `usertype` HAVING states >1
```

addParams()

Sample 1:

1	\$usertype=1;
2	\$model = User::find()
3	->where('usertype = :usertype')
4	->addParams([':usertype' => \$usertype])
5	->one();

Sample 2:

1	\$usertype=1;
2	\$status=0;
3	\$model = User::find()
4	->where('usertype = :usertype and status=:status')
5	->addParams([':usertype' => \$usertype])
6	->addParams([':status' => \$status])
7	// OR Multiple Assigns
8	// ->addParams([':usertype' => \$usertype, ':status' => \$status])
9	->one();

Multiple Conditions

1	\$model = User::find()
2	->where([
3	'type' => 26,
4	'status' => 1,
5	'userid' => [1001,1002,1003,1004,1005],
6])
7	->all();

OUTPUT Query

```
SELECT * FROM `tbl_user` WHERE (`type`=26) AND (`status`=1) AND (`userid` IN (1001, 1002, 1003, 1004, 1005))
```

findBySql

Sample 1:

1	\$sql = 'SELECT * FROM tbl_user';
2	\$model = User::findBySql(\$sql)->all();

Sample 2:

1	\$sql = 'SELECT * FROM tbl_user';
2	\$model = User::findBySql(\$sql)->one();