Query Builder

* Cake\ORM\Query
* https://book.cakephp.org/3.0/en/orm/query-builder.html

The Query Object

* Using Cake\ORM\TableRegistry

$articles = TableRegistry::get(‘Articles’);

// find all

$query = $articles->find();

* Inside ArticlesController.php

$query = $this->Articles->find();

Selecting Rows From A Table

* Find

->find()

* Select

->select([‘colum-1’, ‘colum-2’])

* Distinct

->distinct([‘name’])

* Where
* Where =

->where([‘id’ => 1])

* Where >, <….

->where([‘id >’ => 1]

* And where

->andWhere([‘name’ => ‘npt’]);

* Or where

->orWhere([‘name’ => ‘haha’])

* Example where:
* Ex 1

$query = $articles->find()

->where(['title LIKE' => '%First%'])

->andWhere(function ($exp) {

return $exp->or\_([

'author\_id' => 2,

'is\_highlighted' => true

]);

});

// out

WHERE (

title LIKE '%First%'

AND

(author\_id = 2 OR is\_highlighted = 1)

)

* Ex2:

$query = $articles->find()

->where(function ($exp) {

return $exp

->eq('author\_id', 2)

->eq('published', true)

->notEq('spam', true)

->gt('view\_count', 10);

});

//out

SELECT \*

FROM articles

WHERE (

author\_id = 2

AND published = 1

AND spam != 1

AND view\_count > 10)

* The expression object
* eq()
* notEq()
* like()

$exp->like(‘name’, ‘%A%’)

// WHERE name LIKE "%A%"

* notLike()

$exp->notLike('name', '%A%');

// name NOT LIKE "%A%"

* in()

$exp->in('country\_id', ['AFG', 'USA', 'EST'])

// country\_id IN ('AFG', 'USA', 'EST')

* notIn()
* gt()

$exp->gt('population', '10000')

// population > 10000

* gte()

$exp->gte('population', '10000')

// population >= 10000

* lt()

$exp->lt('population', '10000');

// population < 10000

* lte()

$exp->lte('population', '10000')

// population <= 10000

* isNull()
* isNotNull
* between()

$exp->between('population', 999, 5000000);

// population BETWEEN 999 AND 5000000

* exists()
* notExists()
* Group by

->group(‘name’)

* Having

->having([‘id >’ => 3])

* Order by

->order([‘created’ => ‘DESC’])

* In 3.0.12, addition to order, orderAsc, orderDesc
* Limit

->limit(50)

* Paginate with page 2

->page(2)

* To Array

->toArray();