}

return ['id', 'name', 'address', 'registration\_date'];

public function getOrders()

{

return $this->hasMany(Order::className(), ['customer\_id' => 'id']);

}

}

A primary key of any model can be defined via the primaryKey() method, which defaults to id if not specified. The primary key needs to be placed in the attribute list if you do not manually specify it in the primaryKey() method.

The following is a usage example:

$customer = new Customer();

$customer->name = 'test';

$customer->save();

echo $customer->id; // id will automatically be incremented if not set explicitly // find by query

$customer = Customer::find()->where(['name' => 'test'])->one();

How it works...

The extension provides a Connection component for low-level access to Redis storage records.

You can also use an ActiveRecord-like model with a limited set of methods (where(), limit(), offset(), and indexBy()). Other methods do not exist because Redis does not support SQL queries.

There are no tables in Redis, so you cannot define via relations via a junction table name. You can only define many-to-many relations via other hasMany relations.

For general information on how to use Yii’s ActiveRecord, please refer to Chapter 3, ActiveRecord, Model, and Database.

See also

• For more information about the extension’s usage, see:

° [https ://github.com/yiisoft/yii2-redis/blob/ma stpr/docs/guide/RF ADMF.md](https://github.com/yiisoft/yii2-redis/blob/master/docs/guide/README.md) o [http://www.viiframework.com/doc-2.0/ext-redis-index.html](http://www.yiiframework.com/doc-2.0/ext-redis-index.html)

• For information about Redis key-value storage, refer: <http://redis.io/documentation>

• Chapter 3, ActiveRecord, Model, and Database for ActiveRecord usage