Важно: исполняемый файл yiic должен быть из папки вашего приложения (protected)

**I. Приминение миграции**

1. Для выполнения migrate введите в консоле

yiic migrate up

2. Для отката введите

yiic migrate down (Не все миграции содержат action отката)

**II. Создание новой миграции**

1.создаем файл миграции. Удостоверьтесь в том, что у вас существует папка protected/migrations

и она доступна для записи. Перейдите в консоли в папку protected и запустите команду

php yiic migrate create <name>

2.Действия в миграции.

После прошлого шага у вас в папке protected/migrations появится класс вроде этого:

Class m141013\_082754\_my\_migrate extends CDbMigration

{

public function up()

{

}

public function down()

{

echo "m141013\_082754\_my\_migrate does not support migration down.\n";

return false;

}

}

В классе миграции существует два основных метода: up — выполняется при применении миграции,

а down — в случаи отката. Если откат не возможет, метод down должен вернуть false

Пишем в данном классе, те действия по изменению БД, которые мы хотели сделать.

Доступные методы: execute, insert, update, delete, createTable, renameTable, dropTable, truncateTable,

addColumn, dropColumn, renameColumn, createIndex и другие можно посмотреть в классе CDbMigration: <https://www.yiiframework.com/doc/api/1.1/CDbMigration#insert-detail>

Пример создания таблицы

public function up()

{

$this->createTable('orders', array(

'id' => 'pk',

'user\_id' => 'integer unsigned NOT NULL',

'order\_number' => 'string',

'summa' => 'money',

'status' => 'integer',

'date\_create' => 'datetime',

'last\_status\_change'=> 'datetime',

));

}

public function down()

{

$this->dropTable('orders');

return true;

}

**III. Автоматическая генерация миграций по текущему состоянию БД**

1. Выполняем автоматическую генерацию

yiic initialdbmigration 'name\_of\_your\_database' > initial\_migration.php,

1. Скопируйте из файла initial\_migration методы up и down в заранее созданную миграцию