

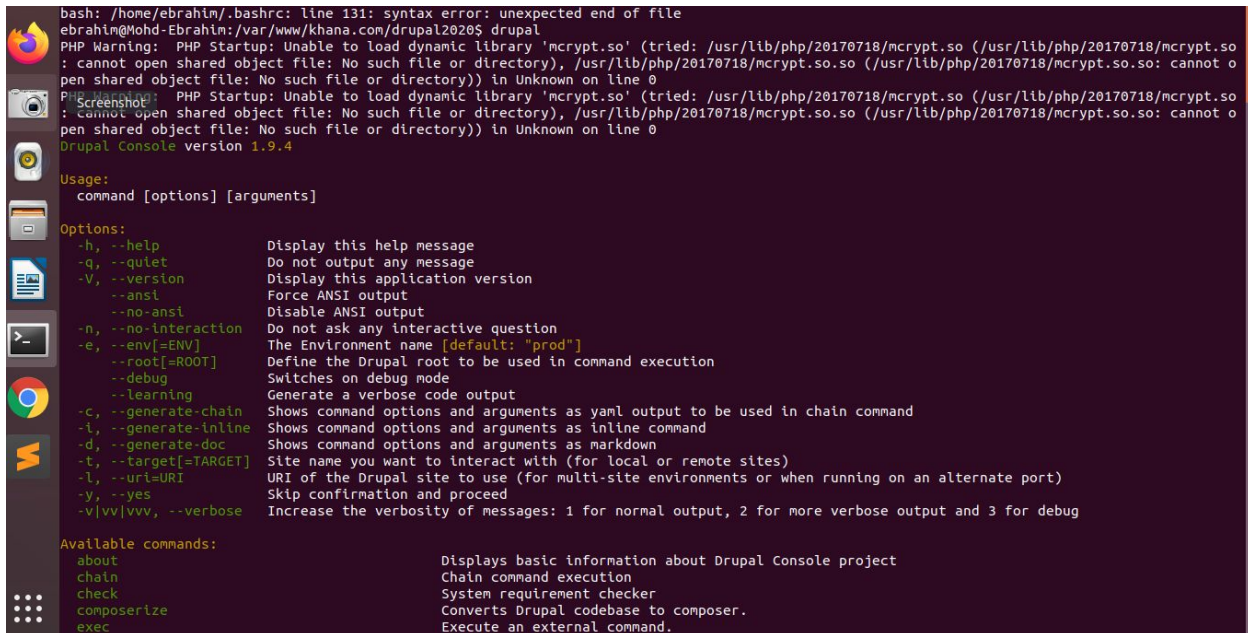
Assignment Custom Entity

Q1. Create a Custom Entity having the following fields :

- Name
- Email
- Contact number

Ans:-

Step1 : install drupal console and type command drupal

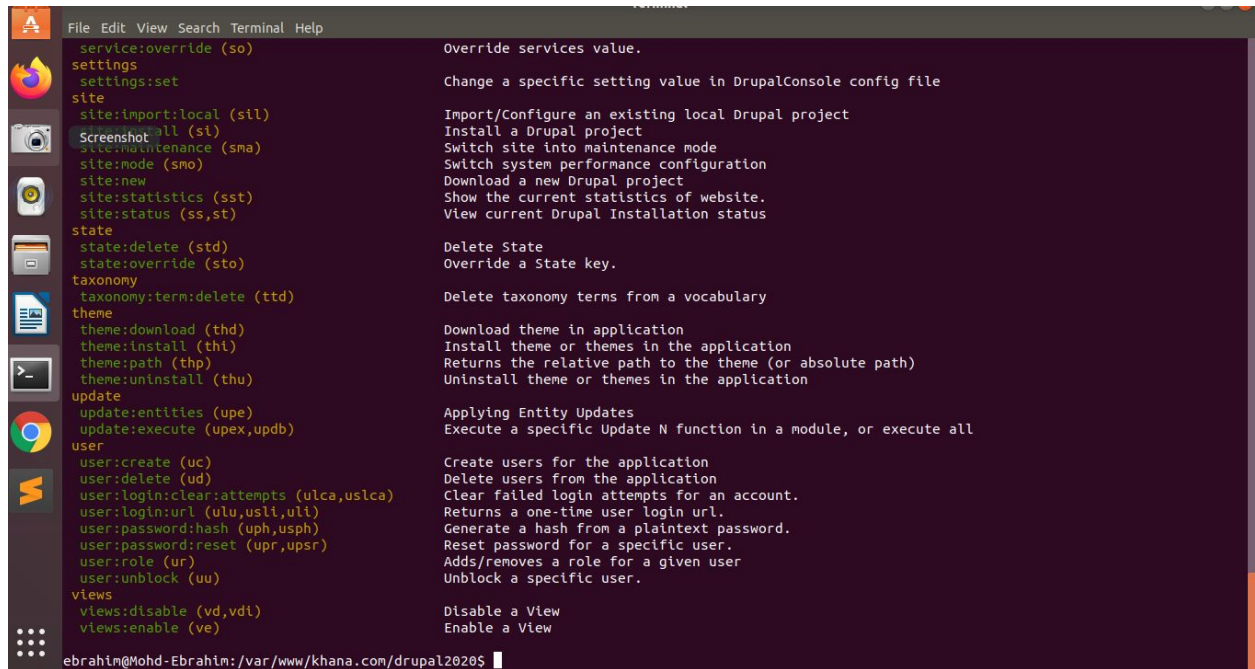


```
bash: /home/ebrahim/.bashrc: line 131: syntax error: unexpected end of file
ebrahim@Mohd-Ebrahim:/var/www/khana.com/drupal2020$ drupal
PHP Warning:  PHP Startup: Unable to load dynamic library 'mcrypt.so' (tried: /usr/lib/php/20170718/mcrypt.so (/usr/lib/php/20170718/mcrypt.so
: cannot open shared object file: No such file or directory), /usr/lib/php/20170718/mcrypt.so.so (/usr/lib/php/20170718/mcrypt.so.so: cannot o
pen shared object file: No such file or directory)) in Unknown on line 0
PHP Warning:  PHP Startup: Unable to load dynamic library 'mcrypt.so' (tried: /usr/lib/php/20170718/mcrypt.so (/usr/lib/php/20170718/mcrypt.so
: cannot open shared object file: No such file or directory), /usr/lib/php/20170718/mcrypt.so.so (/usr/lib/php/20170718/mcrypt.so.so: cannot o
pen shared object file: No such file or directory)) in Unknown on line 0
Drupal Console version 1.9.4

Usage:
  command [options] [arguments]

Options:
  -h, --help                Display this help message
  -q, --quiet               Do not output any message
  -V, --version              Display this application version
                          --ansi Force ANSI output
                          --no-ansi Disable ANSI output
  -n, --no-interaction       Do not ask any interactive question
  -e, --env[=ENV]            The Environment name [default: "prod"]
  --root[=ROOT]              Define the Drupal root to be used in command execution
  --debug                   Switches on debug mode
  --learning                 Generate a verbose code output
  -c, --generate-chain       Shows command options and arguments as yaml output to be used in chain command
  -i, --generate-inline      Shows command options and arguments as inline command
  -d, --generate-doc         Shows command options and arguments as markdown
  -t, --target[=TARGET]      Site name you want to interact with (for local or remote sites)
  -l, --uri=URI              URI of the Drupal site to use (for multi-site environments or when running on an alternate port)
  -y, --yes                  Skip confirmation and proceed
  -v|vv|vvv, --verbose       Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

Available commands:
  about      Displays basic information about Drupal Console project
  chain      Chain command execution
  check      System requirement checker
  composerize Converts Drupal codebase to composer.
  exec       Execute an external command.
```



```
File Edit View Search Terminal Help
service:override (so)      Override services value.
settings
settings:set              Change a specific setting value in DrupalConsole config file
site
site:import:local (sil)    Import/Configure an existing local Drupal project
site:install (si)         Install a Drupal project
site:maintenance (sma)    Switch site into maintenance mode
site:mode (smo)           Switch system performance configuration
site:new                  Download a new Drupal project
site:statistics (sst)     Show the current statistics of website.
site:status (ss,st)       View current Drupal Installation status
state
state:delete (std)        Delete State
state:override (sto)      Override a State key.
taxonomy
taxonomy:term:delete (ttd) Delete taxonomy terms from a vocabulary
theme
theme:download (thd)      Download theme in application
theme:install (thi)       Install theme or themes in the application
theme:path (thp)          Returns the relative path to the theme (or absolute path)
theme:uninstall (thu)     Uninstall theme or themes in the application
update
update:entities (upe)     Applying Entity Updates
update:execute (upex,updb) Execute a specific Update N function in a module, or execute all
user
user:create (uc)          Create users for the application
user:delete (ud)          Delete users from the application
user:login:clear:attempts (ulca,uslca) Clear failed login attempts for an account.
user:login:url (ulu,usli,uli) Returns a one-time user login url.
user:password:hash (uph,usph) Generate a hash from a plaintext password.
user:password:reset (upr,upsr) Reset password for a specific user.
user:role (ur)            Adds/removes a role for a given user
user:unblock (uu)         Unblock a specific user.
views
views:disable (vd,vdi)    Disable a View
views:enable (ve)         Enable a View
ebrahim@Mohd-Ebrahim:/var/www/khana.com/drupal2020$
```

Step2: create custom module using \$ drupal gm command

```
druapl: command not found
ebrahim@Mohd-Ebrahim:/var/www/khana.com/drupal2020$ drupal gm
PHP Warning: PHP Startup: Unable to load dynamic library 'mcrypt.so' (tried: /usr/lib/php/20170718/mcrypt.so (/usr/lib/php/20170718/mcrypt.so
: cannot open shared object file: No such file or directory), /usr/lib/php/20170718/mcrypt.so.so (/usr/lib/php/20170718/mcrypt.so.so: cannot o
pen shared object file: No such file or directory)) in Unknown on line 0
PHP Warning: PHP Startup: Unable to load dynamic library 'mcrypt.so' (tried: /usr/lib/php/20170718/mcrypt.so (/usr/lib/php/20170718/mcrypt.so
: cannot open shared object file: No such file or directory), /usr/lib/php/20170718/mcrypt.so.so (/usr/lib/php/20170718/mcrypt.so.so: cannot o
pen shared object file: No such file or directory)) in Unknown on line 0

// Welcome to the Drupal module generator

Enter the new module name:
> question1

Enter the module machine name [question1]:
>

Enter the module Path [modules/custom]:
>

Enter module description [My Awesome Module]:
> this is my first question

Enter package name [Custom]:
>

Enter Drupal Core version [8.x]:
>

Do you want to generate a .module file? (yes/no) [yes]:
> no

Define module as feature (yes/no) [no]:
> no

Do you want to add a composer.json file to your module? (yes/no) [yes]:
> no

> no
Do you want to generate a unit test class? (yes/no) [yes]:
> no
Do you want to generate a themeable template? (yes/no) [yes]:
> no
Do you want proceed with the operation? (yes/no) [yes]:
>
Generated or updated files
Generation path: /var/www/khana.com/drupal2020
1 - /modules/custom/question1/question1.info.yml

Generated lines: 5
ebrahim@Mohd-Ebrahim:/var/www/khana.com/drupal2020$
```

Step3 : create a custom entity using \$ drupal geco command.

```
Firefox Web Browser
ebrahim@mond-ebrahim: /var/www/khana.com/drupal2020$ drupal geco
PHP Warning: PHP Startup: Unable to load dynamic library 'mcrypt.so' (tried: /usr/lib/php/20170718/mcrypt.so (/usr/lib/php/20170718/mcrypt.so
: cannot open shared object file: No such file or directory), /usr/lib/php/20170718/mcrypt.so.so (/usr/lib/php/20170718/mcrypt.so.so: cannot o
pen shared object file: No such file or directory)) in Unknown on line 0
PHP Warning: PHP Startup: Unable to load dynamic library 'mcrypt.so' (tried: /usr/lib/php/20170718/mcrypt.so (/usr/lib/php/20170718/mcrypt.so
: cannot open shared object file: No such file or directory), /usr/lib/php/20170718/mcrypt.so.so (/usr/lib/php/20170718/mcrypt.so.so: cannot o
pen shared object file: No such file or directory)) in Unknown on line 0

// Welcome to the Drupal Content Entity generator
Enter the module name [admin_toolbar]:
> question1

Enter the class of your new content entity [DefaultEntity]:
> Question

Enter the machine name of your new content entity [question]:
>

Enter the label of your new content entity [Question]:
>

Enter the base-path for the content entity routes [/admin/structure]:
>

Do you want this (content) entity to have bundles? (yes/no) [no]:
> no

Is your entity translatable? (yes/no) [yes]:
> no

Is your entity revisionable? (yes/no) [yes]:
> no

Do you want this (content) entity to have forms? (yes/no) [yes]:
>

Is your entity translatable? (yes/no) [yes]:
> Firefox Web Browser

Is your entity revisionable? (yes/no) [yes]:
> Screenshot

Do you want this (content) entity to have forms? (yes/no) [yes]:
>

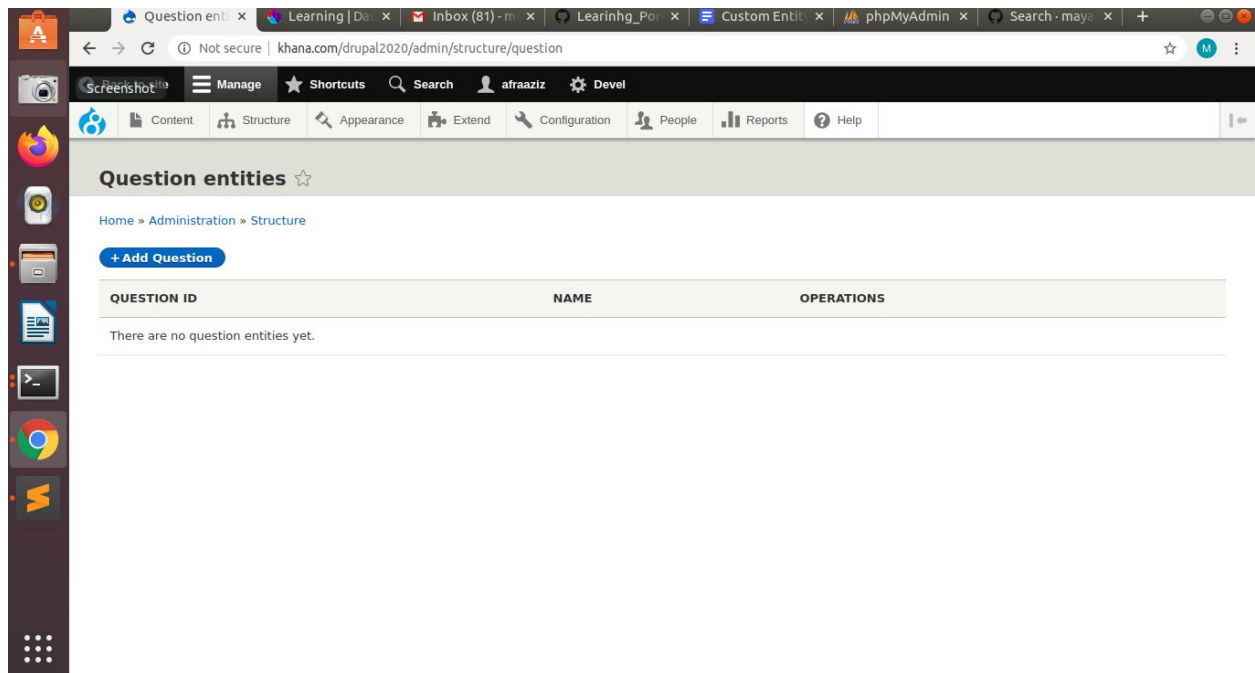
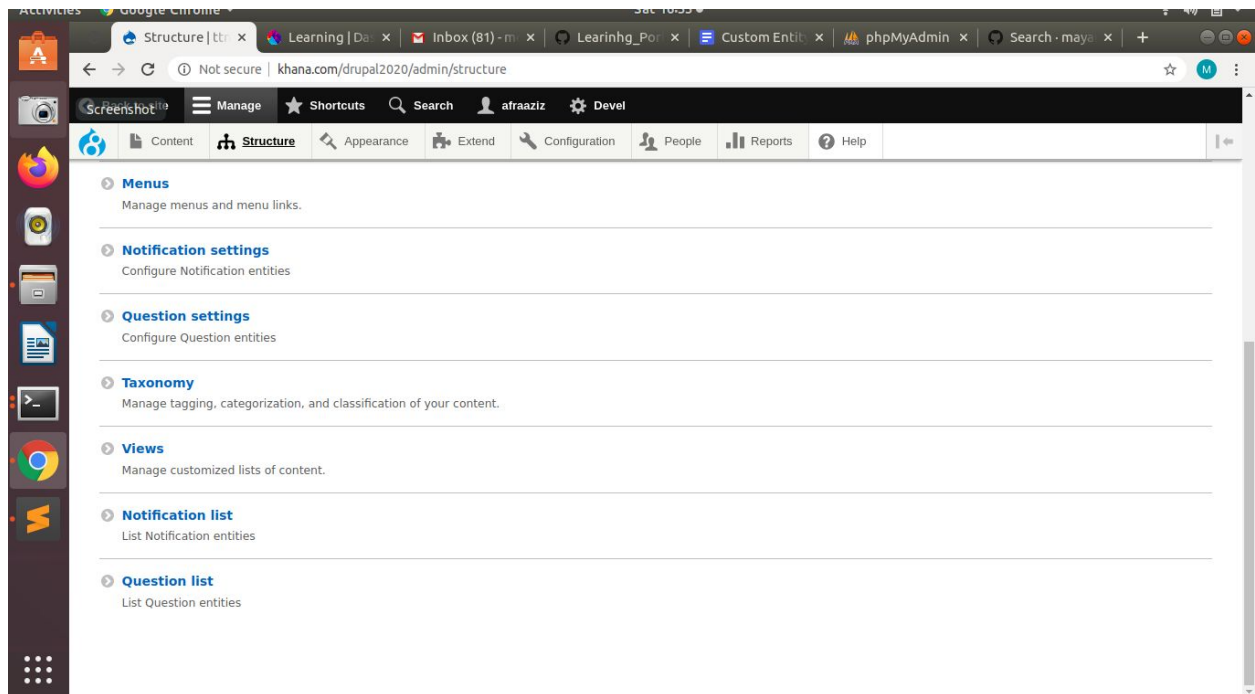
Do you want this (content) entity to have an owner? (yes/no) [yes]:
> no

Generated or updated files
Generation path: /var/www/khana.com/drupal2020
1 - modules/custom/question1/question1.permissions.yml
2 - modules/custom/question1/src/QuestionAccessControlHandler.php
3 - modules/custom/question1/src/Entity/QuestionInterface.php
4 - modules/custom/question1/src/Entity/Question.php
5 - modules/custom/question1/src/Entity/QuestionViewsData.php
6 - modules/custom/question1/src/QuestionListBuilder.php
7 - modules/custom/question1/src/QuestionHtmlRouteProvider.php
8 - modules/custom/question1/question1.links.menu.yml
9 - modules/custom/question1/question1.links.task.yml
10 - modules/custom/question1/question1.links.action.yml
11 - modules/custom/question1/src/Form/QuestionSettingsForm.php
12 - modules/custom/question1/src/Form/QuestionForm.php
13 - modules/custom/question1/src/Form/QuestionDeleteForm.php
14 - modules/custom/question1/templates/question.html.twig
15 - modules/custom/question1/question.page.inc

Generated lines: 614

ebrahim@Mehd-Ebrahim: /var/www/khana.com/drupal2020$
ebrahim@Mehd-Ebrahim: /var/www/khana.com/drupal2020$
```

Now go website and install question1 module and clear all cache.



To add fields email and contact

Go >question1 module>src>Entity>Question.php file and change the following.

```
<?php
```

```
namespace Drupal\question1\Entity;
```

```
use Drupal\Core\Field\BaseFieldDefinition;
```

```
use Drupal\Core\Entity\ContentEntityBase;
```

```
use Drupal\Core\Entity\EntityChangedTrait;
```

```
use Drupal\Core\Entity\EntityPublishedTrait;
```

```
use Drupal\Core\Entity\EntityTypeInterface;
```

```
/**
```

```
 * Defines the Question entity.
```

```
 *
```

```
 * @ingroup question1
```

```
 *
```

```
 * @ContentEntityType(
```

```
 *   id = "question",
```

```
 *   label = @Translation("Question"),
```

```
 *   handlers = {
```

```
 *     "view_builder" = "Drupal\Core\Entity\EntityViewBuilder",
```

```
 *     "list_builder" = "Drupal\question1\QuestionListBuilder",
```

```
 *     "views_data" = "Drupal\question1\Entity\QuestionViewsData",
```

```
 *
```

```
 *     "form" = {
```



```
*   "default" = "Drupal\question1\Form\QuestionForm",
*   "add" = "Drupal\question1\Form\QuestionForm",
*   "edit" = "Drupal\question1\Form\QuestionForm",
*   "delete" = "Drupal\question1\Form\QuestionDeleteForm",
* },
*   "route_provider" = {
*     "html" = "Drupal\question1\QuestionHtmlRouteProvider",
*   },
*   "access" = "Drupal\question1\QuestionAccessControlHandler",
* },
*   base_table = "question",
*   translatable = FALSE,
*   admin_permission = "administer question entities",
*   entity_keys = {
*     "id" = "id",
*     "label" = "name",
*     "uuid" = "uuid",
*     "langcode" = "langcode",
*     "published" = "status",
*   },
*   links = {
*     "canonical" = "/admin/structure/question/{question}",
*     "add-form" = "/admin/structure/question/add",
*     "edit-form" = "/admin/structure/question/{question}/edit",
```

```

*    "delete-form" = "/admin/structure/question/{question}/delete",
*    "collection" = "/admin/structure/question",
* },
* field_ui_base_route = "question.settings"
* )
*/

```

class Question extends ContentEntityBase implements QuestionInterface {

```

    use EntityChangedTrait;

```

```

    use EntityPublishedTrait;

```

```

/**

```

```

 * {@inheritdoc}

```

```

 */

```

```

public function getName() {
    return $this->get('name')->value;
}

```

```

/**

```

```

 * {@inheritdoc}

```

```

 */

```

```

public function setName($name) {
    $this->set('name', $name);
    return $this;
}

```



```
}
```

```
/**
```

```
 * {@inheritdoc}
```

```
 */
```

```
public function getCreatedTime() {  
    return $this->get('created')->value;  
}
```

```
/**
```

```
 * {@inheritdoc}
```

```
 */
```

```
public function setCreatedTime($timestamp) {  
    $this->set('created', $timestamp);  
    return $this;  
}
```

```
/**
```

```
 * {@inheritdoc}
```

```
 */
```

```
public static function baseFieldDefinitions(EntityTypeInterface  
$entity_type) {  
    $fields = parent::baseFieldDefinitions($entity_type);
```

```
// Add the published field.

$fields += static::publishedBaseFieldDefinitions($entity_type);

$fields['name'] = BaseFieldDefinition::create('string')
    ->setLabel(t('Name'))
    ->setDescription(t('The name of the Question entity.'))
    ->setSettings([
        'max_length' => 50,
        'text_processing' => 0,
    ])
    ->setDefaultValue("")
    ->setDisplayOptions('view', [
        'label' => 'above',
        'type' => 'string',
        'weight' => -4,
    ])
    ->setDisplayOptions('form', [
        'type' => 'string_textfield',
        'weight' => -4,
    ])
    ->setDisplayConfigurable('form', TRUE)
    ->setDisplayConfigurable('view', TRUE)
    ->setRequired(TRUE);

//entring email
```

```
$fields['email'] = BaseFieldDefinition::create('string')
->setLabel(t('Email'))
->setDescription(t('The email of the Question entity.'))
->setSettings([
    'max_length' => 50,
    'text_processing' => 0,
])
->setDefaultValue("")
->setDisplayOptions('view', [
    'label' => 'above',
    'type' => 'string',
    'weight' => -4,
])
->setDisplayOptions('form', [
    'type' => 'string_textfield',
    'weight' => -4,
])
->setDisplayConfigurable('form', TRUE)
->setDisplayConfigurable('view', TRUE)
->setRequired(TRUE);

//entring cotact
```

```
$fields['contact'] = BaseFieldDefinition::create('string')
->setLabel(t('Contact'))
->setDescription(t('The contact of the Question entity.'))
->setSettings([
    'max_length' => 50,
    'text_processing' => 0,
])
->setDefaultValue("")
->setDisplayOptions('view', [
    'label' => 'above',
    'type' => 'integer',
    'weight' => -4,
])
->setDisplayOptions('form', [
    'type' => 'string_textfield',
    'weight' => -4,
])
->setDisplayConfigurable('form', TRUE)
->setDisplayConfigurable('view', TRUE)
->setRequired(TRUE);
```

```
$fields['status']->setDescription(t('A boolean indicating whether the  
Question is published.'))
```

```
->setDisplayOptions('form', [  
    'type' => 'boolean_checkbox',  
    'weight' => -3,  
]);
```

```
$fields['created'] = BaseFieldDefinition::create('created')
```

```
->setLabel(t('Created'))
```

```
->setDescription(t('The time that the entity was created.'));
```

```
$fields['changed'] = BaseFieldDefinition::create('changed')
```

```
->setLabel(t('Changed'))
```

```
->setDescription(t('The time that the entity was last edited.'));
```

```
return $fields;
```

```
}
```

```
}
```

Add question ☆

Home » Administration » Structure » Question entities

Name *

The name of the Question entity.

Email *

The email of the Question entity.

Contact *

The contact of the Question entity.

☒ Published

A boolean indicating whether the Question is published.

Save

Showing rows 0 - 1 (2 total, Query took 0.0003 seconds)

`SELECT * FROM `question``

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

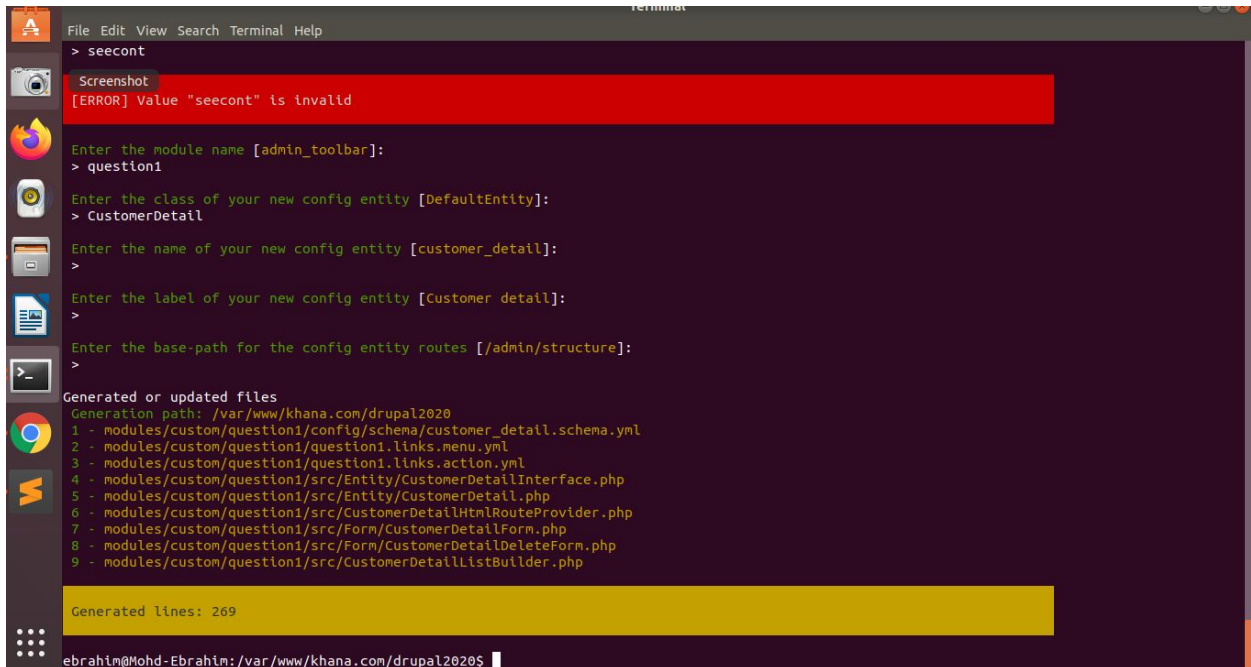
		id	uuid	langcode	status	name	created	changed
<input type="checkbox"/>	Edit Copy Delete	1	a879a7a3-795e-4082-b071-c4274079e244	en	1	Ebrahim	1584185671	1584185671
<input type="checkbox"/>	Edit Copy Delete	2	dfeac81b-2fc2-4139-bd5e-881f6685fce1	en	1	aziz	1584185709	1584185709

Check all | With selected: Edit Copy Delete Export

Q2:- Create a Config Entity to store diff time slots for customers.

Solution:-

Step 1: run the \$ drupal gec command.



```
File Edit View Search Terminal Help
> seecont
Screenshot
[ERROR] Value "seecont" is invalid

Enter the module name [admin_toolbar]:
> question1

Enter the class of your new config entity [DefaultEntity]:
> CustomerDetail

Enter the name of your new config entity [customer_detail]:
>

Enter the label of your new config entity [Customer detail]:
>

Enter the base-path for the config entity routes [/admin/structure]:
>

Generated or updated files
Generation path: /var/www/khana.com/drupal2020
1 - modules/custom/question1/config/schema/customer_detail.schema.yml
2 - modules/custom/question1/question1.links.menu.yml
3 - modules/custom/question1/question1.links.action.yml
4 - modules/custom/question1/src/Entity/CustomerDetailInterface.php
5 - modules/custom/question1/src/Entity/CustomerDetail.php
6 - modules/custom/question1/src/Entity/CustomerDetailHtmlRouteProvider.php
7 - modules/custom/question1/src/Form/CustomerDetailForm.php
8 - modules/custom/question1/src/Form/CustomerDetailDeleteForm.php
9 - modules/custom/question1/src/Controller/CustomerDetailListBuilder.php

Generated lines: 269
ebrahim@Mohd-Ebrahim: /var/www/khana.com/drupal2020$
```

Step 2: go to>>module>>custom>>your module>>src>>Form>>CustomerDetailForm.php and do changes the following

```
<?php
```

```
namespace Drupal\question1\Form;
```

```
use Drupal\Core\Entity\EntityForm;
```

```
use Drupal\Core\Form\FormStateInterface;
```

```
/**
```

```
 * Class CustomerDetailForm.
```

```
 */
```

```
class CustomerDetailForm extends EntityForm {
```



```

/**
 * {@inheritdoc}
 */
public function form(array $form, FormStateInterface $form_state) {
    $form = parent::form($form, $form_state);

    $customer_detail = $this->entity;
    $form['label'] = [
        '#type' => 'textfield',
        '#title' => $this->t('Label'),
        '#maxlength' => 255,
        '#default_value' => $customer_detail->label(),
        '#description' => $this->t("Label for the Customer detail."),
        '#required' => TRUE,
    ];

    $form['id'] = [
        '#type' => 'machine_name',
        '#default_value' => $customer_detail->id(),
        '#machine_name' => [
            'exists' => '\Drupal\question1\Entity\CustomerDetail::load',
        ],
        '#disabled' => !$customer_detail->isNew(),
    ];
}

```

```
];
```

```
/* You will need additional form elements for your custom properties. */
```

```
$form['timeslot'] = [  
  '#type' => 'select',  
  '#options'=>[1=>'Time Slot -1(9:00 am - 11:00am)',2=>'Time Slot  
-2(11:00 am - 1:00pm)',3=>'Time Slot -3(1:00 pm - 3:00pm)'],  
  '#title' => 'TimeSlot',  
  //' #default_value'=>$customer_detail->timeslot(),  
  //' #machine_name' => [  
    //' #exists' => '\Drupal\question1\Entity\CustomerDetail::load',  
  //],  
  //' #disabled' => !$customer_detail->isNew(),  
];
```

```
return $form;  
}
```

```
/**
```

```
 * {@inheritdoc}
```

```
 */
```

```
public function save(array $form, FormStateInterface $form_state) {
```

```

$customer_detail = $this->entity;
$status = $customer_detail->save();

switch ($status) {
    case SAVED_NEW:
        $this->messenger()->addMessage($this->t('Created the %label
Customer detail.', [
            '%label' => $customer_detail->label(),
        ]));
        break;

    default:
        $this->messenger()->addMessage($this->t('Saved the %label
Customer detail.', [
            '%label' => $customer_detail->label(),
        ]));
    }
    $form_state->setRedirectUrl($customer_detail->toUrl('collection'));
}

}

```

Now go to >>config>>schema and do changes the following
question1.customer_detail.*:

type: config_entity

label: 'Customer detail config'

mapping:

id:

type: string

label: 'ID'

label:

type: label

label: 'Label'

uuid:

type: string

timeslot:

type: string

label: 'Timeslot'

