Django 2.1

MODEL FIELD REFERENCE

INSTRUCTOR: JORGE GARCÍA CÁRDENAS

EMAIL: JOR@MISENA.EDU.CO

Django 2.1

MODELO DE REFERENCIA DE CAMPOS O TIPO DE DATOS

INSTRUCTOR: JORGE GARCÍA CÁRDENAS

EMAIL: JOR@MISENA.EDU.CO

Django 2.1 - Model Field Reference

- Tipos de Campos (los más comunes)
- Opciones de Campos
- Campos de relación
  - 1 : 1
  - 1 : N
  - M : N
- Documentación Referencia de la API de Campos
- Referencia de atributos de Campos

Django 2.1 - Model Field Reference

- Tipos de Campos (los más comunes)

AutoField(**options)

Entero que incrementa su valor de 1 en 1 de acuerdo al ID. No es necesario especificarlos, se agrega automáticamente al modelo.

BigAutoField (**options)

Entero de 64 bits. (1 a 9’’’223.372’’036.854’775.807)

- *Ejemplo options:*

primary_key=True

Implica → null=False and unique=True.

```
from django.db import models

# Create your models here.
# Tablas: clientes y vehículos

class Cliente(models.Model):
    idCliente = models.BigAutoField(primary_key=True)
    nombre = models.CharField(max_length=254)
    apellido = models.CharField(max_length=254)
```

Django 2.1 - Model Field Reference

- Tipos de Campos (los más comunes)

SmallIntegerField(**options)

Entero pequeño. (-32.768 to 32.767)

IntegerField(**options)

Entero. (-2.147’483.648 a 2.147’483.647)

BigIntegerField(**options)

Entero de 64 bits. (-9223372036854775808 a 9’’’223.372’’036.854’775.807)

- *Ejemplo options:*

null = True

blank = True

default = 0

editable = False . Para no mostrarlo en el admin, no se toma en cuenta para la validación.

primary_key = True

unique = True

Django 2.1 - Model Field Reference

- Tipos de Campos (los más comunes)

FloatField(**options)

Número de coma flotante.

Es una instancia **Float** de Python.

DecimalField(max_digits=None, decimal_places=None, **options)

Número decimal de precisión fija.

Es una instancia **Decimal** de Python.

- *Ejemplo options:*

null = True

blank = True

default = 0

editable = False

- *Ejemplo options: para representar 999.00*

max_digits=5

decimal_places=2

Django 2.1 - Model Field Reference

- Tipos de Campos (los más comunes)

CharField(max_length=None, **options)

Cadena de caracteres. Por compatibilidad use hasta el máximo estándar 254 caracteres.

EmailField(max_length=254, **options)

```
class ClienteDeportes(models.Model):
    DEPORTES_CHOICES = (
        ('SOF', 'Sófbol'),
        ('FUT', 'Fútbol'),
        ('NAT', 'Natación'),
        ('TEN', 'Tenis'),
    )
    deportePractica = models.CharField(max_length=3, choices=DEPORTES_CHOICES, default='NAT')
    frecuencia = models.IntegerField()
```

TextField(**options)

Campo de texto grande. El widget de formulario predeterminado para este campo es un Textarea (HTML).

La opción *max_length* permite controlar caracteres en el formulario pero no en base de datos.

-

Ejemplo options:

null = True

blank = True

default = ''

editable = False

primary_key = True

unique = True

choices

Django 2.1 - Model Field Reference

- Tipos de Campos (los más comunes)

DateField(auto_now=False, auto_now_add=False, **options)

Fecha, instancia Python de *datetime.date*.

DateTimeField(auto_now=False, auto_now_add=False, **options)

Fecha y hora.

TimeField (auto_now=False, auto_now_add=False, **options)

Hora, instancia Python de *datetime.time*.

- *Ejemplo options:*

null = True

blank = True

default = 0

editable = False

help_text = "Por favor usar el siguiente formato: YYYY-MM-DD."

```
fechaHoraCreado = models.DateTimeField(auto_now_add=True)
fechaHoraActualizado = models.DateTimeField(auto_now=True)
```

Django 2.1 - Model Field Reference

- Tipos de Campos (los más comunes)

FileField(upload_to=None, max_length=100, **options)

Un campo de carga de archivos.

```
class MyModel(models.Model):
    # file will be uploaded to MEDIA_ROOT/uploads
    upload = models.FileField(upload_to='uploads/')
    # or...
    # file will be saved to MEDIA_ROOT/uploads/2015/01/30
    upload = models.FileField(upload_to='uploads/%Y/%m/%d/')
```

ImageField(upload_to=None, height_field=None, width_field=None, max_length=100, **options)

Valida que el objeto cargado sea una imagen.

Hereda de FileField.

Nota: Requiere la librería Pillow.

- *Ejemplo options:*

null = True

blank = True

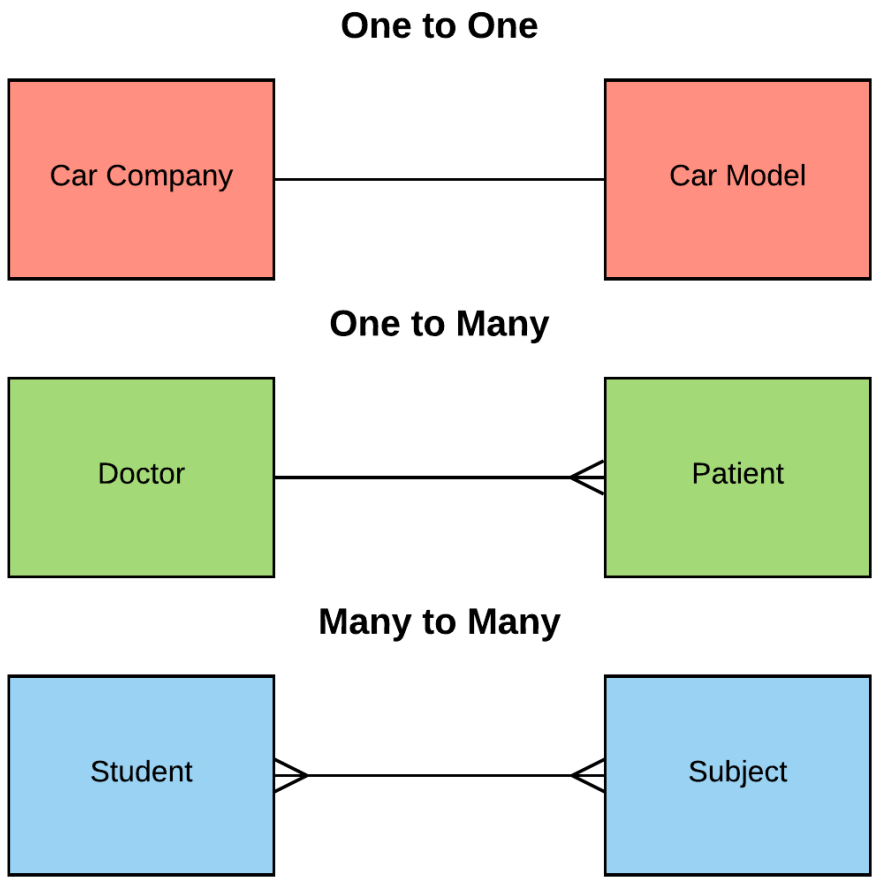
default = ‘’

```
class Car(models.Model):
    name = models.CharField(max_length=255)
    price = models.DecimalField(max_digits=5, decimal_places=2)
    photo = models.ImageField(upload_to='cars')
```

```
>>> car = Car.objects.get(name="57 Chevy")
>>> car.photo
<ImageFieldFile: chevy.jpg>
>>> car.photo.name
'cars/chevy.jpg'
>>> car.photo.path
'/media/cars/chevy.jpg'
>>> car.photo.url
'http://media.example.com/cars/chevy.jpg'
```

Django 2.1 - Model Field Reference

- Campos de relación
  - 1 : 1 (OneToOneField)
  - 1 : N (ForeignKeyField)
  - M : N (ManyToManyField)



Django 2.1 - Model Field Reference

- Campos de relación
  - 1 : 1 (OneToOneField)
  - 1 : N (ForeignKeyField)
  - M : N (ManyToManyField)

Types of Relationships

- One to one
  - Person to Driver's License, Country to President, Person to Social Security Number (in theory)
- One to many
  - Mother to Child, Division to Department, Supervisor to Employees, Owner to Car, Customer to Invoice
- Many to many
  - Parent to child, student to professor, nurse to patient, Invoice to Inventory Part, student to class

Imagen tomada de: https://slideplayer.com/slide/7379133/

Django 2.1 - Model Field Reference

- Campos de relación
  - 1 : 1 (OneToOneField)

```
class PresidentePais(models.Model):
    usuario = models.OneToOneField(Pais, on_delete=models.CASCADE,)
    nombre = models.CharField(max_length=254)
    apellido = models.CharField(max_length=254)
```

Django 2.1 - Model Field Reference

- Campos de relación
  - 1 : N (ForeignKey)
    - Una relación de muchos a uno. Requiere dos argumentos posicionales: la clase con la que se relaciona el modelo y la opción ***on_delete***.
      - **CASCADE**: borrado en cascada
      - PROTECT: Previene la eliminación del objeto al que se hace referencia. Concepto de Clase-Objeto-Subclase.
      - SET_NULL: Establecer el ForeignKey null; esto solo es posible si null = True.
      - SET_DEFAULT: Establecer el ForeignKey a su valor por defecto; debe definir valor por defecto default=‘’.
      - SET: Set the ForeignKey to the value passed to SET()
      - **DO_NOTHING**: No tomar ninguna medida. Si el backend de su base de datos impone integridad referencial, esto causará un IntegrityError a menos que agregue manualmente una restricción de SQL ON DELETE al campo de la base de datos.

Django 2.1 - Model Field Reference

- Campos de relación
  - 1 : N (ForeignKey)
    - Para crear una relación recursiva, un objeto que tiene una relación de varios a uno consigo mismo, use:
      - models.ForeignKey('self', on_delete=models.CASCADE).

```
class Usuario(models.Model):
    jefe = models.ForeignKey('self', on_delete=models.CASCADE)
    nombre = models.CharField(max_length=254)
    apellido = models.CharField(max_length=254)
```

    - Si necesita crear una relación en un modelo que aún no se ha definido, puede usar el nombre del modelo, en lugar del propio objeto modelo:

```
class Car(models.Model):
    manufacturer = models.ForeignKey(
        'Manufacturer',
        on_delete=models.CASCADE,
    )
    # ...

class Manufacturer(models.Model):
    # ...
    pass
```

Django 2.1 - Model Field Reference

- Campos de relación
  - M : N (ManyToManyField)

  Una relación de muchos a muchos. Requiere un argumento posicional: la clase con la que se relaciona el modelo, que funciona exactamente igual que para ForeignKey, incluidas las relaciones recursivas y perezosas.

  Representación de base de datos

  Detrás de escena, Django crea una tabla de unión intermedia para representar la relación de muchos a muchos. De forma predeterminada, este nombre de tabla se genera utilizando el nombre del campo muchos a muchos y el nombre de la tabla para el modelo que lo contiene.

  Debido a que algunas bases de datos no admiten nombres de tablas de cierta longitud, estos nombres de tablas se truncarán automáticamente a 64 caracteres y se utilizará un hash de exclusividad. Esto significa que puede ver nombres de tablas como **author_books_9cdf4**; esto es perfectamente normal Puede proporcionar manualmente el nombre de la tabla de unión utilizando la opción **db_table**.

Django 2.1 - Model Field Reference

- Campos de relación
  - M : N (ManyToManyField)

```python
class Person(models.Model):
    name = models.CharField(max_length=50)

class Group(models.Model):
    name = models.CharField(max_length=128)
    members = models.ManyToManyField(
        Person,
        through='Membership',
        through_fields=('group', 'person'),
    )

class Membership(models.Model):
    group = models.ForeignKey(Group, on_delete=models.CASCADE)
    person = models.ForeignKey(Person, on_delete=models.CASCADE)
    inviter = models.ForeignKey(
        Person,
        on_delete=models.CASCADE,
        related_name="membership_invites",
    )
    invite_reason = models.CharField(max_length=64)
```

Django 2.1 - Model Field Reference

- Documentación Referencia de la API de Campos
- Referencia de atributos de Campos

Para más información ir a la Documentación de Django 2.1.x

https://docs.djangoproject.com/en/2.1/ref/models/fields/

Muchas gracias!

Django 2.1

MODEL FIELD REFERENCE

INSTRUCTOR: JORGE GARCÍA CÁRDENAS

EMAIL: JOR@MISENA.EDU.CO