


Django Forms com muitos dados

Jamil Atta Junior
atta.jamil@gmail.com
<http://github.com/jamilatta>

Pythonbrasil[9] 

SciELO

- +440.000 artigos científicos
- +1.000 periódicos
- +10.000.000 citações
- +1.800.000 acessos/dia

A comemoração dos 15 anos da Rede SciELO reunirá autoridades e especialistas em pesquisa e comunicação científica. O objetivo do encontro é destacar e debater o estado da arte em comunicação científica em acesso aberto e os desafios para o desenvolvimento dos periódicos científicos e do Programa SciELO.



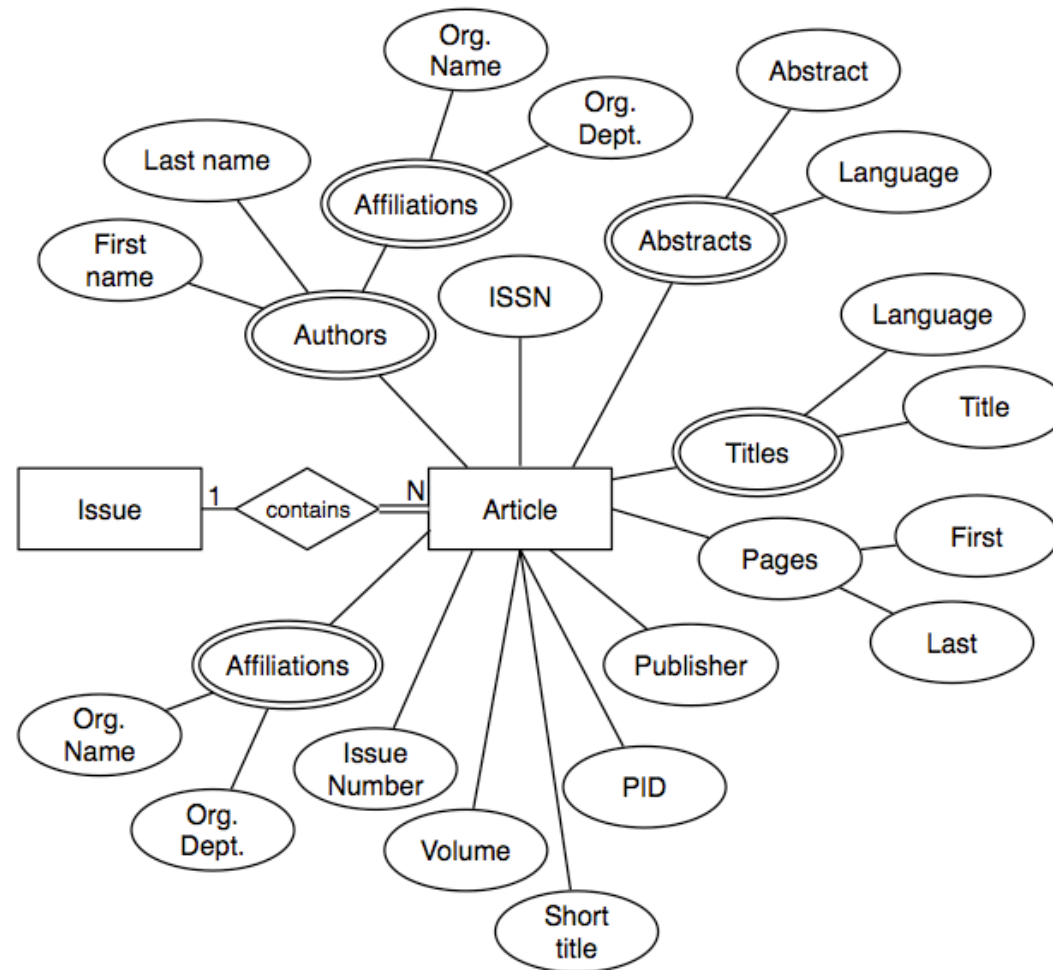
<http://www.scielo15.org/>

Jamil Atta Junior
atta.jamil@gmail.com
<http://github.com/jamilatta>

Pythonbrasil[9]



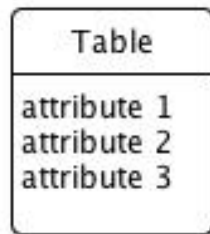
Dados semi-estruturados



<http://www.slideshare.net/gustavofons/modelagem-de-dados-semiestruturados-com-isisdm>

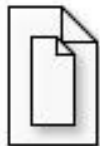
Para construir formulários... com o que se preocupar?

Crescimento dos dados



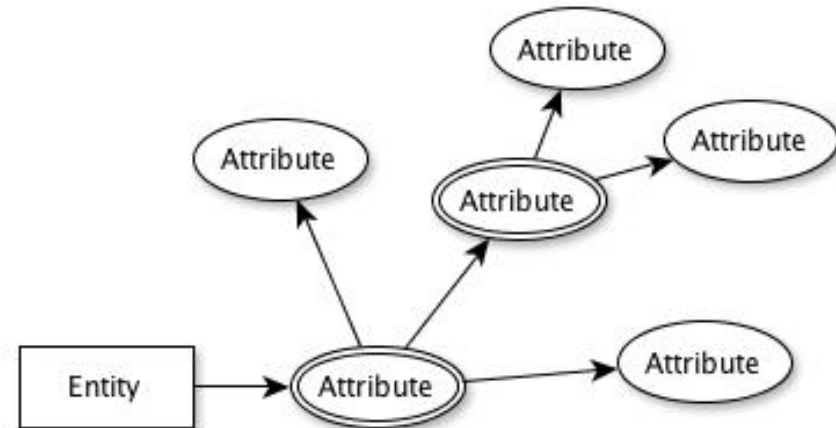
Tabelas que representam atributos multivalorados

Formulários alinhados



Jamil Atta Junior
atta.jamil@gmail.com
<http://github.com/jamilatta>

Modelagem dos dados

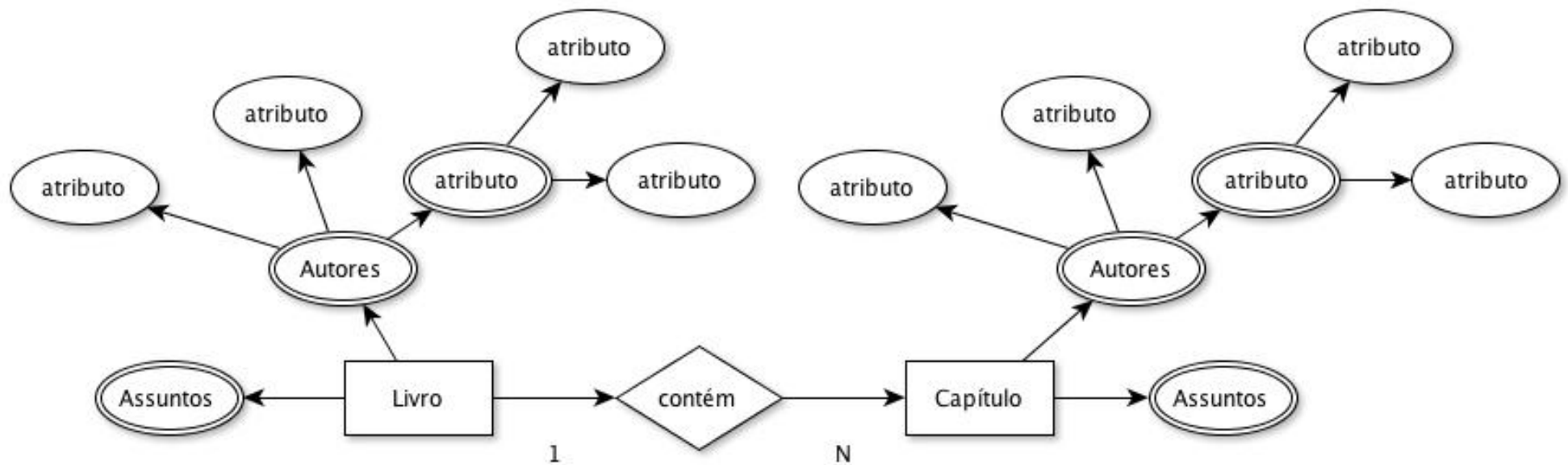


Campos que geram lentidão no carregamento do formulário

- ForeignKey
- ManyToManyField
- Inline Formset (seus relacionamento)



Exemplo livro capítulo



Evitar o uso de formtsets, inlineformsets e utilizar AJAX para os campos ForeignKey e ManyToManyField

Como evitar lentidão no carregamento dos formulários

- Override queryset nos objetos do formulário
- AJAX para os campos de relacionamentos
- Componentes/Campos customizados
- Uso de métodos `only()` e `defer()`

Campos customizados no Django

Subclasse do campo ModelMultipleChoiceField (fields.py)

```
class CustomField(forms.ModelMultipleChoiceField):  
    widget = CustomWidget  
  
    def __init__(self, model, *args, **kwargs):  
        #Necessário informar o modelo  
        #Nesse momento é possível utilizar ``queries``  
        mais baratas.
```

Subclasse widget TextInput (widgets.py)

```
class CustomWidget(forms.TextInput):  
  
    class Media:  
        css = {"all": ("css/.css",)}  
        js = ("js.js",)  
  
    def __init__(self, attrs=None, **kwargs):  
  
    def render(self, name, value, attrs=None):  
        #Sobreescreva o método render
```

<https://docs.djangoproject.com/en/1.5/howto/custom-model-fields/>

select_related e prefetch_related

- . Sempre explicitando os campos de relacionamento
- . Utilize o param **depth** para limitar lookups
- . Use prefetch para ManyToManyField
- . Uso do atributo db_indexed=True

Cacheamento

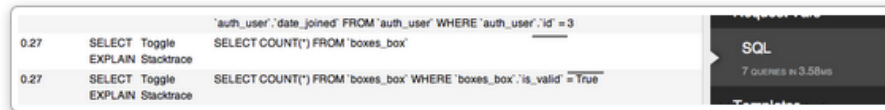
Diminui o tempo de resposta.

No Django há um grande suporte a memcached.

Opção de criar “custom backend”.

Ferramentas para analisar a performance

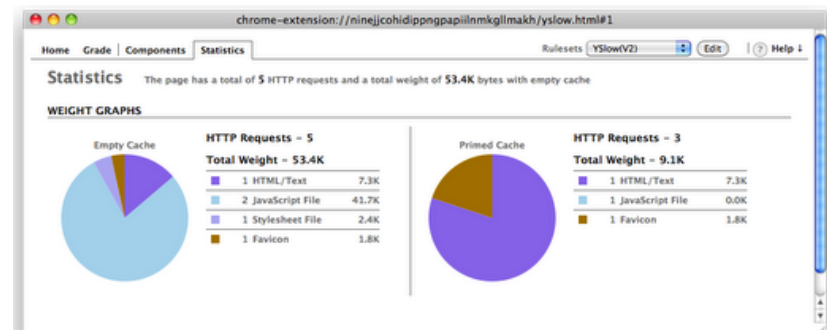
- Django Debug Toolbar (cache_panel.CachePanel)
- Yslow (extensão para o Firefox e Chrome)
- Django Timelog (similar ao Ngnix Logging)
- Ferramentas de Load Testing (ab, FunkLoad, Jmeter e HTTPPerf)
- NewRelic (application monitoring)



Debug Toolbar

view	method	status	count	minimum	maximum	mean	stdev
bookcap.views.add_book	POST	302	1	0.200	0.200	0.200	0
bookcap.views.list_book	GET	500	1	0.120	0.120	0.120	0
bookcap.views.add_book	GET	200	24	1.840	8.300	6.638	5.430

Timelog



Yslow

Jamil Atta Junior
atta.jamil@gmail.com
<http://github.com/jamilatta>

Pythonbrasil[9]



Dúvidas???

Obrigado.

<http://www.slideshare.net/JamilAttaJunior>

Jamil Atta Junior
atta.jamil@gmail.com
<http://github.com/jamilatta>

Pythonbrasil[9]

