

# Django: QuerySets e ORM



guilherme-lima

27/01/2020

## Django: QuerySets e ORM

O mapeador de objeto-relacional [ORM](#) do **Django** é compatível com MySQL, PostgreSQL, SQLite e Oracle.

## O que o ORM do Django faz?

O ORM do Django é fornecido com construções especiais de abstração que podem ser usadas para criar consultas complexas ao banco de dados. Em outras palavras, no lugar de realizar uma ação direta no banco de dados com código SQL por exemplo, utilizamos o ORM como ponte de comunicação entre o banco e a aplicação.

## O que é um QuerySet?

**QuerySet** é um conjunto de ações que serão realizadas no banco de dados, ou seja, podemos criar, buscar, atualizar ou deletar os dados sem escrever a query SQL será executada no banco.

Abaixo, segue uma lista com as principais ações de um QuerySet:

## Ações de um QuerySet

Usaremos como base o seguinte modelo, usado no curso [Fundamentos de Django da Alura](#):

```
from django.db import models

class Pessoa(models.Model):
    nome = models.CharField(max_length=200)
    email = models.CharField(max_length=200)

    def __str__(self):
        return self.nome
```

Podemos também criar uma instância de um objeto na memória e posteriormente salvar no banco de dados usando o método `save`:

```
peessoa = Pessoa(nome='Adriélly', email='adrielly@alura.com')
peessoa.save()
```

A ação anterior, executa o **insert no banco de dados**. Vimos como criar um objeto na memória primeiro e depois persistir no banco de dados, mas também podemos criar o objeto e persistir no banco de dados em uma única operação usando o método `create`:

```
Pessoa.objects.create(nome='Guilherme', email='gui@alura.com')
```

Já para buscar todas as pessoas cadastradas no banco, usamos o método `all`:

```
Pessoa.objects.all()
```

Caso queira buscar uma instância do banco de dados que contenha o nome *Guilherme* por exemplo, posso utilizar o método `get`:

```
peessoa = Pessoa.objects.get(nome='Guilherme')
```

O método `get` permite recuperar um **único objeto do banco de dados** (por exemplo, uma `primary key`). Se nenhum resultado for retornado pelo banco de dados, esse método gerará uma exceção de **DoesNotExist** indicando que o objeto não existe e, se o banco de dados retornar mais de um resultado, gerará uma exceção **MultipleObjectsReturned**, indicando que existe múltiplos objetos retornados.

Para atualizar o email do Guilherme por exemplo, utilizamos também o método `save`, porém, dessa vez, **o método executa uma instrução SQL UPDATE**, já que temos uma instância.

```
peessoa.email = 'guilherme@alura.com'
peessoa.save()
```

As alterações feitas no objeto não são persistidas no banco de dados até que o método `save` seja executado.

Você pode excluir determinados resultados do seu `QuerySet` usando o método `delete`:

```
peessoa = Pessoa.objects.get(id=1)
peessoa.delete()
```

## Filtros

Para os próximos exemplos, considere o seguinte modelo:

```
from django.db import models
from datetime import datetime

class Receita(models.Model):
    nome_receita = models.CharField(max_length=200)
    modo_preparo = models.TextField()
    date_receita = models.DateTimeField(default=datetime.now, blank=True)
    publicada = models.BooleanField(default=False)

    def __str__(self):
        return self.nome_receita
```

Para filtrar um `QuerySet`, você pode usar o método `filter`. Por exemplo, podemos recuperar todas as receitas publicadas, como mostra o código abaixo:

```
receitas_publicadas = Receita.objects.filter(publicada=True)
```

É possível também filtrar por vários campos:

```
receita_publicada_chocolate = Receita.objects.filter(publicada=True, no
```

## Ordenação

Também é possível recuperar todos os objetos ordenados com o método `order_by`, por exemplo, todas as receitas ordenadas pelo nome:

```
receitas_ordenadas = Receita.objects.order_by('nome_receita')
```

A ordem crescente está **implícita**. Você pode indicar a ordem decrescente com um prefixo de sinal negativo, assim:

```
receitas_ordenadas = Receita.objects.order_by('-nome_receita')
```

## O que aprendemos neste artigo

Ao invés de escrever o código SQL, podemos pedir de uma forma elegante para o ORM do Django realizar essa ação por nós. Você também pode concatenar quantos filtros desejar para um QuerySet e não atingirá o banco de dados até que o QuerySet seja avaliado. Todos os exemplos mostrados acima, foram executados nos [cursos de Django da Alura](#).

Cursos de Programação

## Quer mergulhar em tecnologia e aprendizagem?

Receba a newsletter que o nosso CEO escreve pessoalmente, com insights do mercado de trabalho, ciência e desenvolvimento de software

Escreva seu email

[ME INSCREVA](#)

## TAMBÉM EM ALURA

<b>Trabalhando com precisão em números ...</b>	<b>Utilizando export.modules no ...</b>	<b>Django: instalação, configuração e ...</b>	<b>Aprenda 10 anos</b>
É bem possível que você já tenha se deparado com pequenos erros de precisão com números float. Entenda ...	Saiba como exportamos módulos no Node.js	Este artigo te ensinará o que é e como trabalhar com o framework Django, um dos mais usados para ...	Começar a ser rápido, tecnologia levar anos



**0 Comentários** **Alura**  **Disqus' Privacy Policy** **Entrar** ▼ **Recomendar**  **Tweet**  **Compartilhar****Ordenar por Mais votados** ▼

Iniciar a discussão...

FAZER LOGIN COM

OU REGISTRE-SE NO DISQUS 

Seja o primeiro a comentar.

 **Inscreve-se**  **Adicione o Disqus no seu site** **Adicionar Disqus** **Adicionar**  **Do Not Sell My Data**

ARTIGOS &gt; PROGRAMAÇÃO

**Cursos profissionais de  
Programação é na Alura,  
comece agora!**

PREMIUM	PREMIUM PLUS	MAX
✓	1250 cursos ✓	✓
✓	Certificado de participação ✓	✓
✓	App para Android e iPhone/iPad ✓	✓
✓	Projeto avaliado pelos instrutores ✓	✓
✓	Acesso à Alura Start ✓	✓
✓	Acesso à Alura Língua ✓	✓
12X <b>R\$75</b>	12X <b>R\$100</b>	12X <b>R\$120</b>
à vista R\$900	à vista R\$1.200	à vista R\$1.440
MATRICULE-SE	MATRICULE-SE	MATRICULE-SE

Conheça os **Planos para Empresas** →

## NAVEGAÇÃO

PLANOS

TODOS OS CURSOS

GUIAS DE EMPRESAS  
INSTRUTORES

COMO VIRAR INSTRUTOR

ARTIGOS

PARA ESCOLAS

IMERSÕES

PODCASTS

POLÍTICA DE PRIVACIDADE

TERMOS DE USO

SOBRE NÓS

COMO FUNCIONA

DEV EM <T>

PERGUNTAS FREQUENTES

STATUS

#### NOSSAS REDES E APPS



#### FALE COM A GENTE

WHATSAPP

OUTROS CANAIS

#### EMPRESA/GOVERNO?

Planos Corporativos Para A Sua Equipe.

ALURA PARA EMPRESAS

## CURSOS

Cursos de Mobile

[Flutter, React Native](#) | [iOS e Swift](#) | [Android, Kotlin](#) | [Jogos](#)

## **Cursos de Programação**

[Lógica](#) | [Python](#) | [PHP](#) | [Java](#) | [.NET](#) | [Node JS](#) | [Ruby & Rails](#) | [C](#) | [Computação](#) | [Jogos](#)

## **Cursos de Front-end**

[HTML, CSS](#) | [React](#) | [Angular](#) | [JavaScript](#) | [jQuery](#)

## **Cursos de DevOps**

[Segurança](#) | [Cloud](#) | [Linux](#) | [IoT](#) | [Redes](#)

## **Cursos de Design & UX**

[Photoshop e Illustrator](#) | [Usabilidade e UX](#) | [Vídeo e Motion](#) | [3D](#)

## **Cursos de Marketing Digital**

[Ads](#) | [Marketing](#) | [SEO](#) | [E-commerce](#) | [Social Media](#) | [YouTube](#)

## **Cursos de Data Science**

[SQL e Banco de Dados](#) | [NoSQL](#) | [Data Science](#) | [Machine Learning](#) | [BI](#) | [Estatística](#) | [Excel](#)

## **Cursos de Inovação & Gestão**

[Métodos Ágeis](#) | [Softskills](#) | [Liderança e Gestão](#) | [Startups](#) | [Vendas](#)