

# Table of Contents

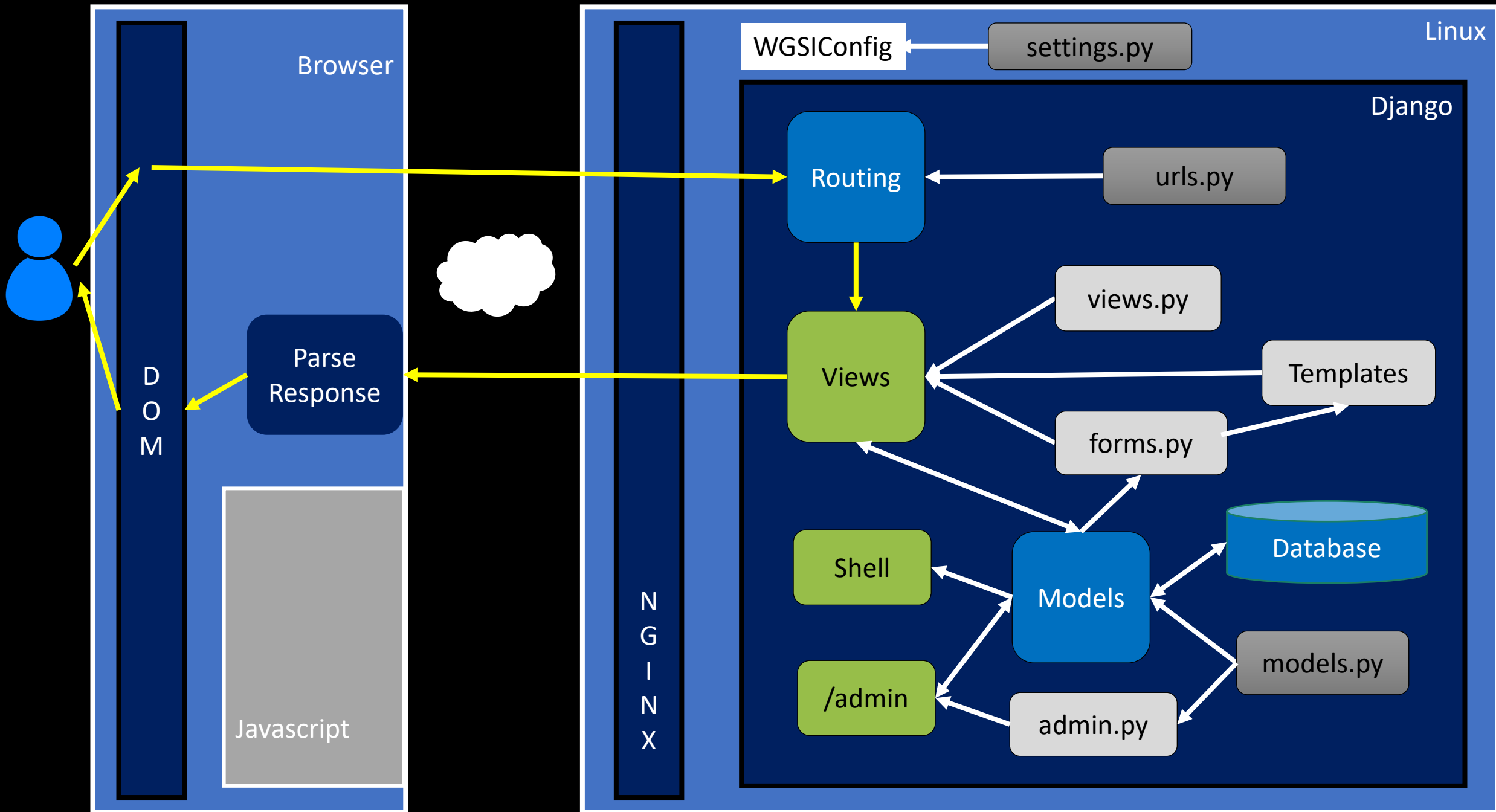
This slide deck consists of slides used in 5 lecture videos in Week 4. Below is a list of shortcut hyperlinks for you to jump into specific sections.

- (page 2) [Week 4: One-to-Many Models Overview](#)
- (page 10) [Week 4: Removing Replication in One-to-Many Models](#)
- (page 18) [Week 4: Storing Primary and Foreign Keys in a Database](#)
- (page 24) [Week 4: Representing One-To-Many MModels in Django](#)
- (page 32) [Week 4: Using the Django Shell to Explore ONE-to-Many Models](#)

Charles Severance  
[www.dj4e.com](http://www.dj4e.com)

# Data Modelling

## One to Many



# Model Design

# Model Design

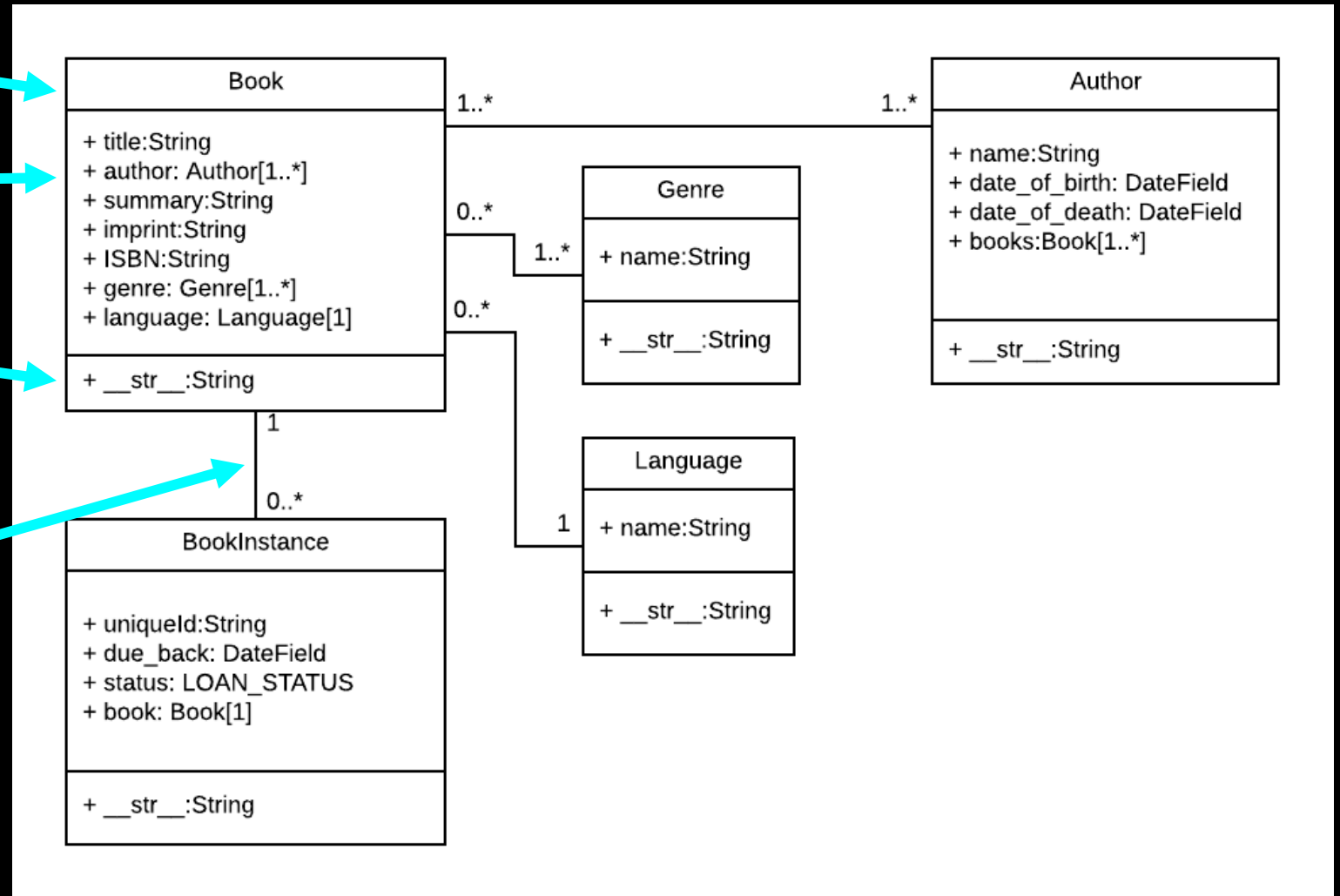
- Model design is an **art form** of its own with particular skills and experience
- Our goal is to avoid the really bad mistakes and design clean and easily understood models
- Model design starts with a sample data set and draws a picture

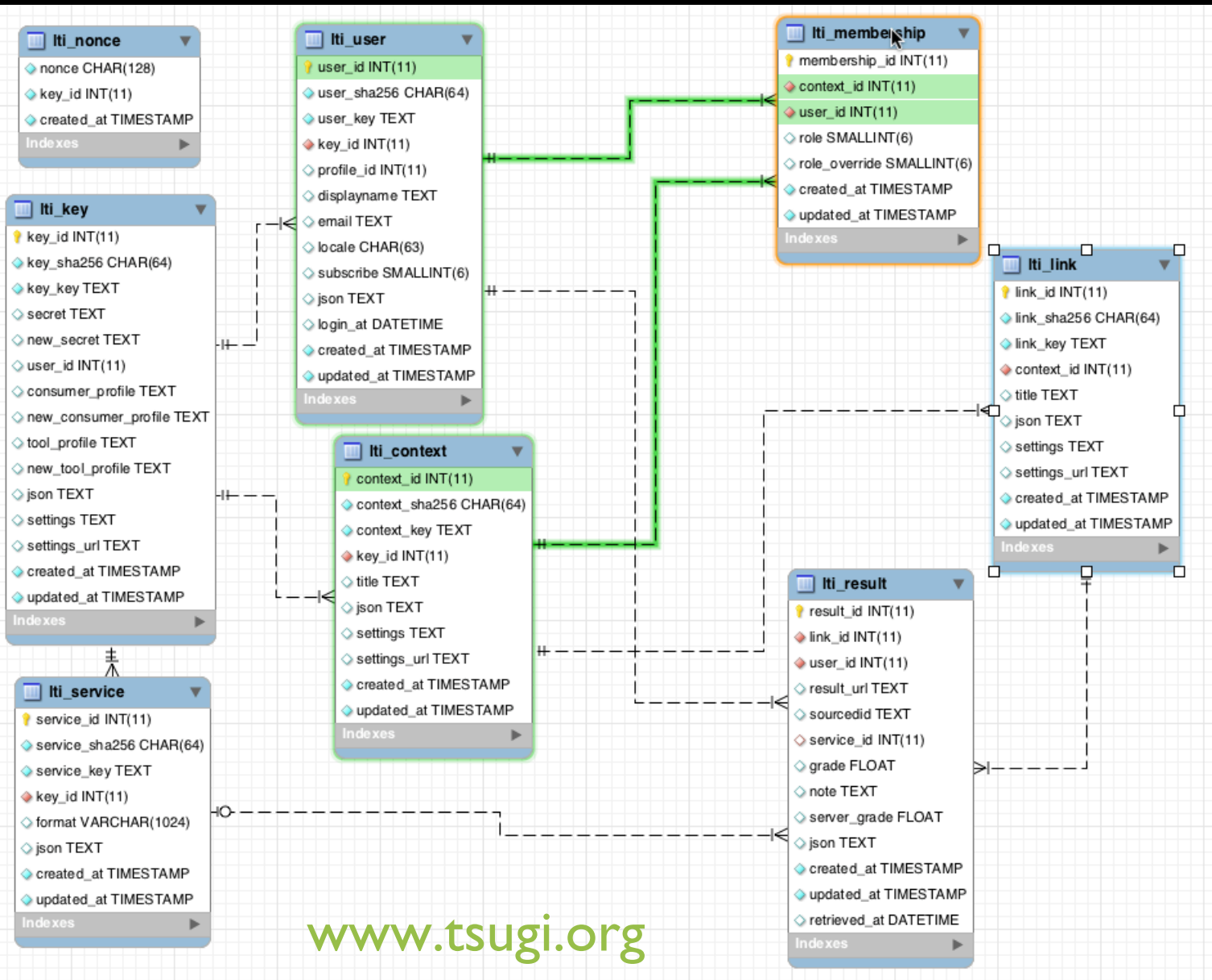
Table

Field

Method

"One Book can have  
between zero and infinite  
number of BookInstances"









# Database Normalization (3NF)

- There is \*tons\* of database theory / math – we simplify this to a few rules...
- **Do not replicate data** - reference data - point at data
- Add a special “**unique key**” column to each table which we will make references to. By convention, many programmers and frameworks call this column “**id**”
- Use **integers** for to make links between tables – integers are fast and small

[http://en.wikipedia.org/wiki/Database\\_normalization](http://en.wikipedia.org/wiki/Database_normalization)

# Designing a Data Model



Title	ISBN	Instance	Due_back	Status	Language	Genre	Author
Wisdom of Crowds	385721706	1		Available	en	Think	James Surowiecki
Wisdom of Crowds	385721706	2	next week	On Loan	en	Think	James Surowiecki
Wisdom of Crowds	385721706	3	who knows	On Loan	en	Think	James Surowiecki
Introduction to Networking	9781511654944	1		Available	en	Tech	Charles Severance
Introducción a las Redes	9781523627516	1		Available	es	Tech	Fernando Tardio
Raspberry PI	9781624311390	1		Available	en	Tech	Kristen Fontichiaro
Where Wizards Stay Up Late	0684812010	1		Available	en	Tech	Katy Hafner
Innovators Dilemma	0066620694	1		Available	en	Think	Clayton Christensen
Unlocking the Clubhouse	0262133989	1		Available	en	Think	Ann Margolis
Mindshift	9781101982853	1		Available	en	Think	Barbara Oakley
Python for Everybody	9781530051120	1		Available	en	Tech	Charles Severance
Python per tutti	1730907164	1		Available	it	Tech	Vittore Zen
Weaving the Web	00625187X	1		Available	en	Tech	Tim Berners-Lee

Title	ISBN	Instance	Due_back	Status	Language	Genre	Author
Wisdom of Crowds	385721706	1		Available	en	Think	James Surowiecki
Wisdom of Crowds	385721706	2	next week	On Loan	en	Think	James Surowiecki
Wisdom of Crowds	385721706	3	who knows	On Loan	en	Think	James Surowiecki
Introduction to Networking	9781511654944	1		Available	en	Tech	Charles Severance
Introducción a las Redes	9781523627516	1		Available	es	Tech	Fernando Tardio
Raspberry PI	9781624311390	1		Available	en	Tech	Kristen Fontichiaro
Where Wizards Stay Up Late	0684812010	1		Available	en	Tech	Katy Hafner
Innovators Dilemma	0066620694	1		Available	en	Think	Clayton Christensen
Unlocking the Clubhouse	0262133989	1		Available	en	Think	Ann Margolis
Mindshift	9781101982853	1		Available	en	Think	Barbara Oakley
Python for Everybody	9781530051120	1		Available	en	Tech	Charles Severance
Python per tutti	1730907164	1		Available	it	Tech	Vittore Zen
Weaving the Web	00625187X	1		Available	en	Tech	Tim Berners-Lee

# Removing Duplication

Title	ISBN	Instance	Due_back	Status	Language
Wisdom of Crowds	385721706	1		Available	en
Wisdom of Crowds	385721706	2	next week	On Loan	en
Wisdom of Crowds	385721706	3	who knows	On Loan	en
Introduction to Networking	9781511654944	1		Available	en
Introducción a las Redes	9781523627516	1		Available	es

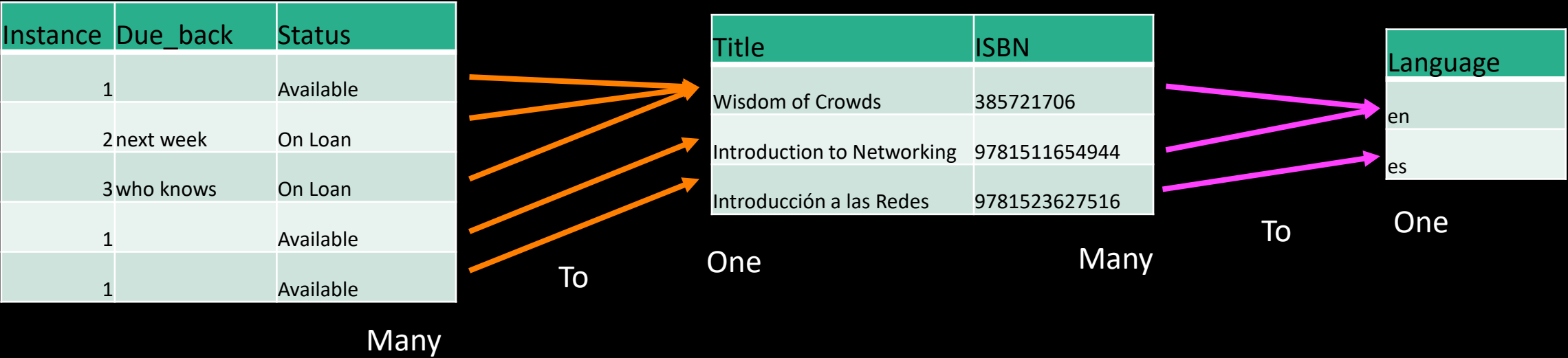
Instance	Due_back
1	
2	next week
3	who knows
1	
1	

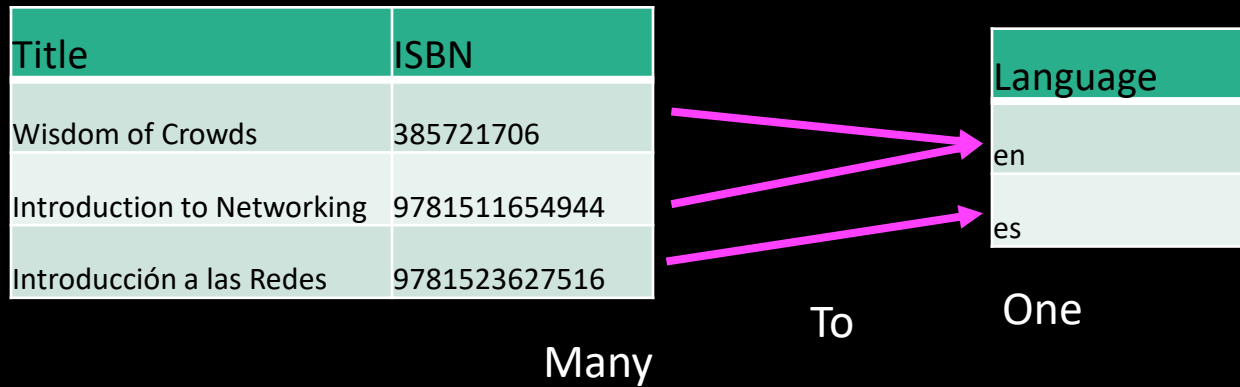
Title	ISBN
Wisdom of Crowds	385721706
Introduction to Networking	9781511654944
Introducción a las Redes	9781523627516

Language
en
es

Title	ISBN	Instance	Due_back	Status	Language
Wisdom of Crowds	385721706	1		Available	en
Wisdom of Crowds	385721706	2	next week	On Loan	en
Wisdom of Crowds	385721706	3	who knows	On Loan	en
Introduction to Networking	9781511654944	1		Available	en
Introducción a las Redes	9781523627516	1		Available	es

# Adding Links



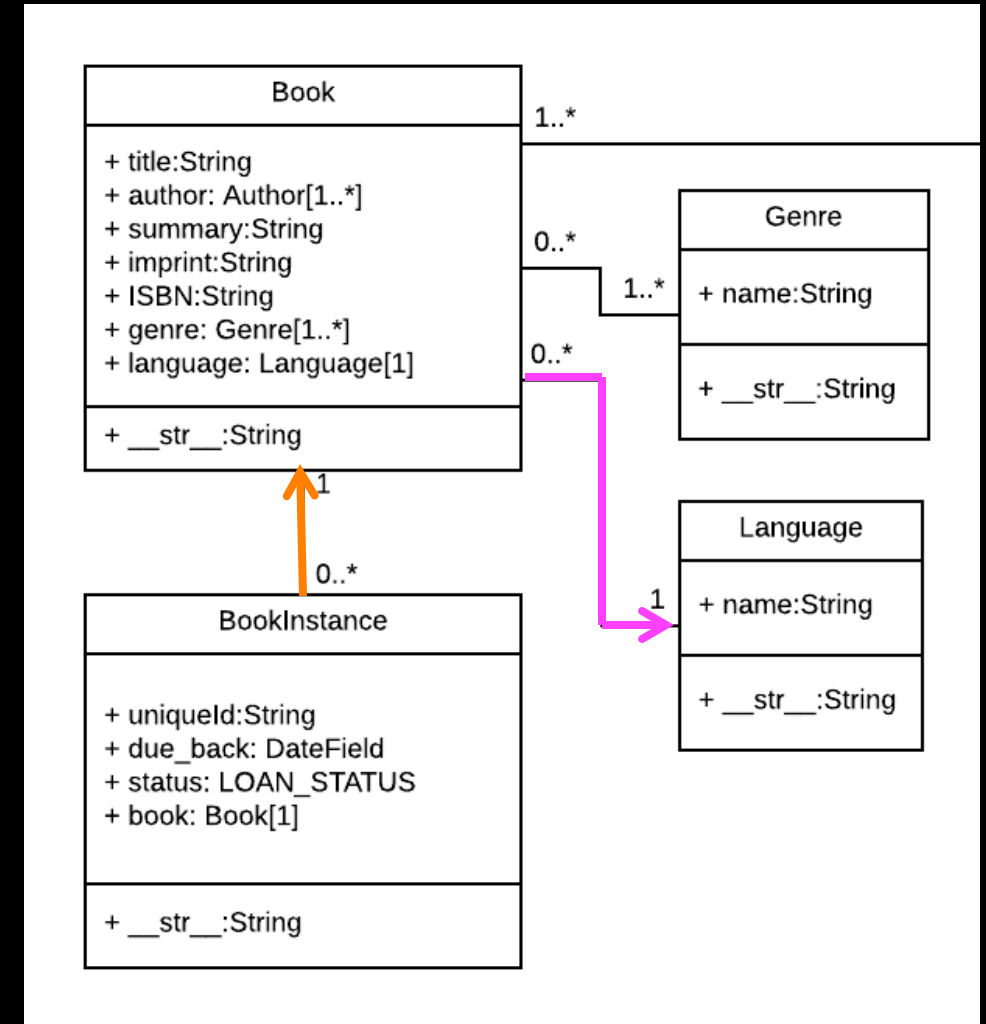


Legend:

1 One

1..\* Many with a minimum of 1

0..\* Many with a minimum of 0





Instance	Due_back	Status
1		Available
2	next week	On Loan
3	who knows	On Loan
1		Available
1		Available

Many



To

Title	ISBN
Wisdom of Crowds	385721706
Introduction to Networking	9781511654944
Introducción a las Redes	9781523627516

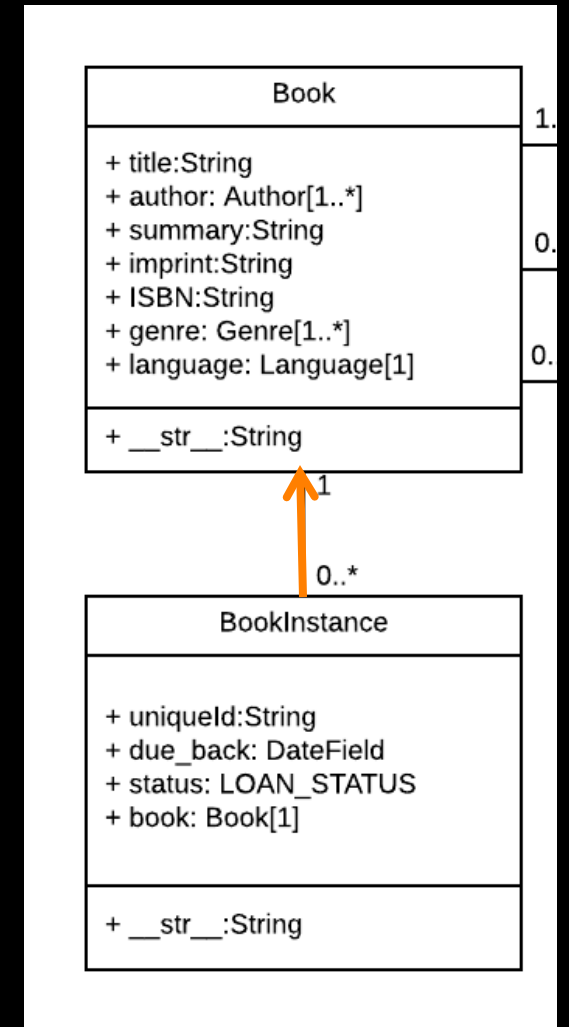
One

Legend:

1 One

1..\* Many with a minimum of 1

0..\* Many with a minimum of 0

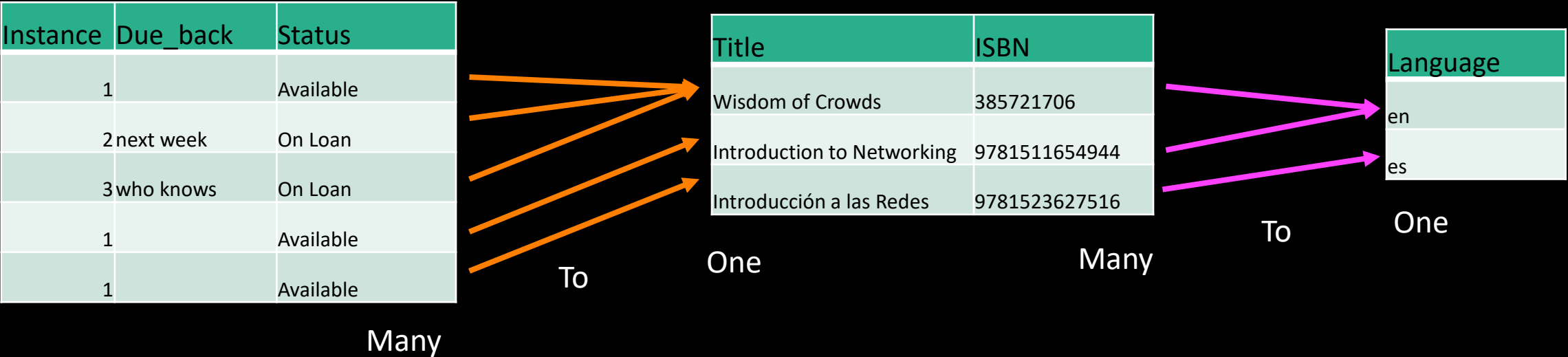


# Representing Links (Relationships) in a Database

Lets get physical...

Title	ISBN	Instance	Due_back	Status	Language
Wisdom of Crowds	385721706	1		Available	en
Wisdom of Crowds	385721706	2	next week	On Loan	en
Wisdom of Crowds	385721706	3	who knows	On Loan	en
Introduction to Networking	9781511654944	1		Available	en
Introducción a las Redes	9781523627516	1		Available	es

# Links in a Logical Model



Title	ISBN	Instance	Due_back	Status	Language
Wisdom of Crowds	385721706	1		Available	en
Wisdom of Crowds	385721706	2	next week	On Loan	en
Wisdom of Crowds	385721706	3	who knows	On Loan	en
Introduction to Networking	9781511654944	1		Available	en
Introducción a las Redes	9781523627516	1		Available	es

## Links in a Physical Model

id	Instance	Due_back	book_id
1	1		1
2	2	next week	1
3	3	who knows	1
4	1		2
5	1		3

id	Title	ISBN	lang_id
1	Wisdom of Crowds	385721706	1
2	Introduction to Networking	9781511654944	1
3	Introducción a las Redes	9781523627516	2

id	Lang
1	en
2	es



Title	ISBN	Instance	Due_back	Status	Language
Wisdom of Crowds	385721706	1		Available	en
Wisdom of Crowds	385721706	2	next week	On Loan	en
Wisdom of Crowds	385721706	3	who knows	On Loan	en
Introduction to Networking	9781511654944	1		Available	en
Introducción a las Redes	9781523627516	1		Available	es

# Links in a Physical Model

id	Instance	Due_back	book_id
1	1		1
2	2	next week	1
3	3	who knows	1
4	1		2
5	1		2

id	Title	ISBN	lang_id
1	Wisdom of Crowds	385721706	1
2	Introduction to Networking	9781511654944	1
3	Introducción a las Redes	9781523627516	2

id	Lang
1	en
2	es

# Key Terminology

- We add an *automatically incrementing* column to every row which we call the "**Primary Key**" for that row. We often name the column "**id**" to indicate that it is the "identifier" for that row.
- When we add a column to a table that "points to" a row in another table we call it a "**Foreign Key**" and often include the name of the destination table in the column name like "**lang\_id**"

id	Title	ISBN	lang_id
1	Wisdom of Crowds	385721706	1
2	Introduction to Networking	9781511654944	1
3	Introducción a las Redes	9781523627516	2



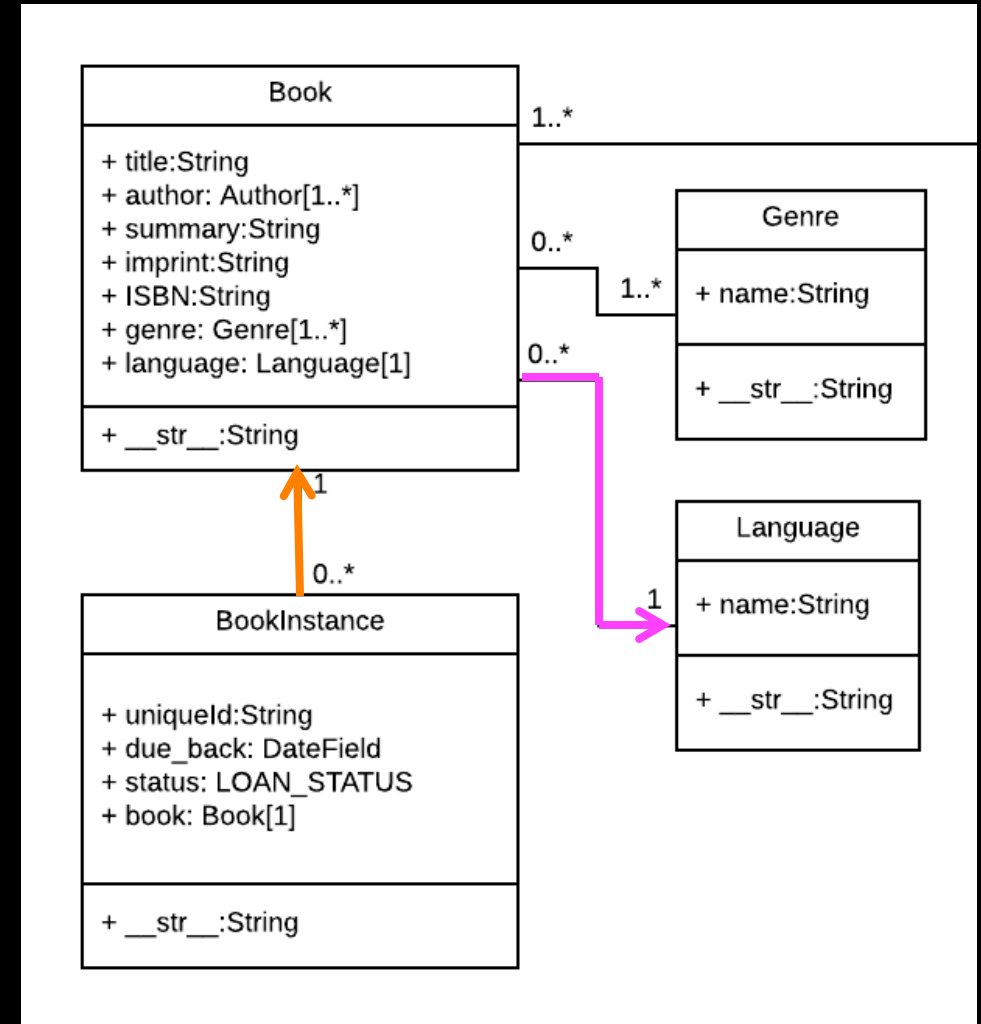
id	Lang
1	en
2	es

# Physical / Logical

id	Instance	Due_back	book_id
1	1		1
2		2next week	1
3		3who knows	1
4	1		2
5	1		2

id	Lang
1	en
2	es

id	Title	ISBN	lang_id
1	Wisdom of Crowds	385721706	1
2	Introduction to Networking	9781511654944	1
3	Introducción a las Redes	9781523627516	2



# Representing Links (Relationships) in Django

Lets get our ORM on...



# Model Field Types

- AutoField
- BigAutoField
- BigIntegerField
- BinaryField
- BooleanField
- CharField
- DateField
- DateTimeField
- DecimalField
- DurationField
- EmailField
- FileField
- FilePathField
- FloatField
- ImageField
- IntegerField
- GenericIPAddressField
- NullBooleanField
- PositiveIntegerField
- PositiveSmallIntegerField
- SlugField
- SmallIntegerField
- TextField
- TimeField
- URLField
- **ForeignKey**
- **ManyToManyField**
- **OneToOneField**

<https://docs.djangoproject.com/en/3.0/ref/models/fields/#field-types>

Instance	Due_back	Status
1		Available
2	next week	On Loan
3	who knows	On Loan

Title	ISBN
Wisdom of Crowds	385721706
Introduction to Networking	9781511654944
Introducción a las Redes	9781523627516

Language
en
es

```

from django.db import models

class Lang(models.Model):
    name = models.CharField(max_length=200)

class Book(models.Model):
    title = models.CharField(max_length=200)
    isbn = models.CharField(max_length=13)
    lang = models.ForeignKey('Lang', on_delete=models.SET_NULL, null=True)

class Instance(models.Model):
    book = models.ForeignKey('Book', on_delete=models.CASCADE)
    due_back = models.DateField(null=True, blank=True)

```

```

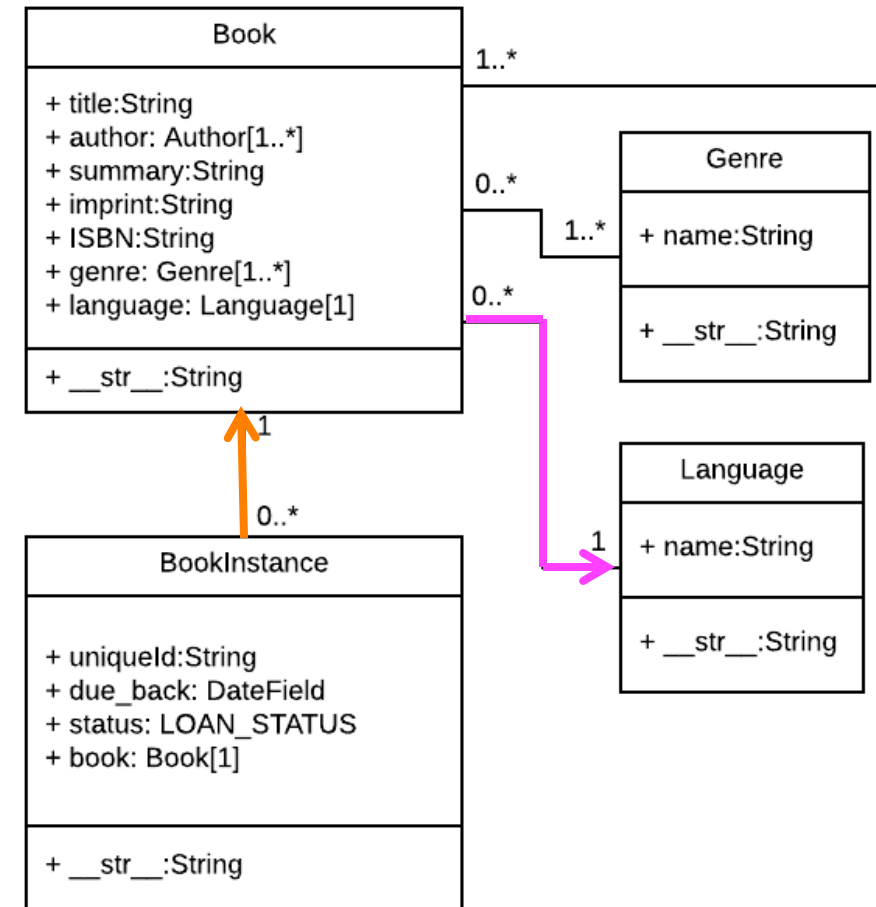
from django.db import models

class Lang(models.Model):
    name = models.CharField(max_length=200)

class Book(models.Model):
    title = models.CharField(max_length=200)
    isbn = models.CharField(max_length=13)
    lang = models.ForeignKey('Lang',
                             on_delete=models.SET_NULL, null=True)

class Instance(models.Model):
    due_back = models.DateField(null=True, blank=True)
    book = models.ForeignKey('Book',
                             on_delete=models.CASCADE)

```



<https://github.com/csev/dj4e-samples/blob/master/bookone/models.py>

# From Model to Database

```
$ python3 manage.py makemigrations
```

```
Migrations for 'bookone':
```

```
bookone/migrations/0001_initial.py
```

- Create model Book
- Create model Instance
- Create model Lang
- Add field lang to book

```
$ python3 manage.py migrate
```

```
Operations to perform:
```

```
Apply all migrations: admin,...
```

```
Running migrations:
```

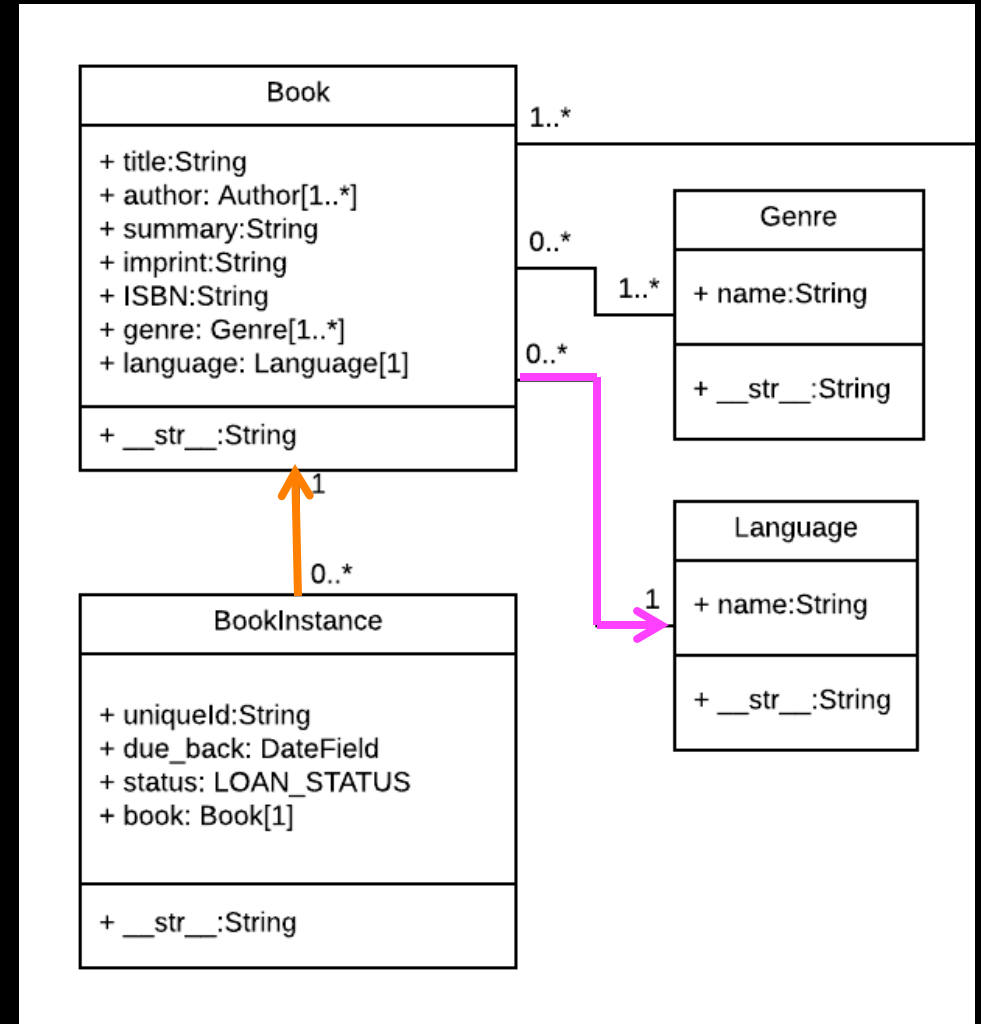
```
Applying bookone.0001_initial... OK
```

Note that makemigrations only "does something" when you create or alter a models.py file. The migrate only "does something" when there are migrations that are not yet applied to the database. Also an application must be added to settings.py before these commands see the models.py file for an application.

```

dj4e-samples$ sqlite3 db.sqlite3
SQLite version 3.24.0 2018-06-04 14:10:15
Enter ".help" for usage hints.
sqlite> .tables 'bookone%'
bookone_book  bookone_instance bookone_lang
sqlite> .schema bookone_book
CREATE TABLE IF NOT EXISTS "bookone_book" (
  "id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
  "title" varchar(200) NOT NULL,
  "isbn" varchar(13) NOT NULL,
  "lang_id" integer NULL REFERENCES "bookone_lang" ("id")
  DEFERRABLE INITIALLY DEFERRED
);
CREATE INDEX "bookone_book_lang_id_24ba3759"
  ON "bookone_book" ("lang_id");
sqlite> .schema bookone_lang
CREATE TABLE IF NOT EXISTS "bookone_lang" (
  "id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
  "name" varchar(200) NOT NULL
);
sqlite> .schema bookone_instance
CREATE TABLE IF NOT EXISTS "bookone_instance" (
  "id" integer NOT NULL PRIMARY KEY AUTOINCREMENT,
  "due_back" date NULL,
  "book_id" integer NOT NULL REFERENCES "bookone_book" ("id")
  DEFERRABLE INITIALLY DEFERRED
);
CREATE INDEX "bookone_instance_book_id_1fa5e2e7"
  ON "bookone_instance" ("book_id");
sqlite> .quit
dj4e-samples$

```

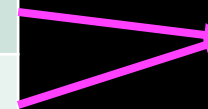


# About on\_delete

- What do we do when a row in one table points to a row in a "foreign" table via a foreign key and the "destination row" is deleted
  - on\_delete = set\_null – Keep the row but set foreign key to null
  - on\_delete = cascade - Delete the row

id	Title	ISBN	lang_id
1	Wisdom of Crowds	385721706	1
2	Introduction to Networking	9781511654944	1
3	Introducción a las Redes	9781523627516	2

id	Lang
1	en
2	es



[https://docs.djangoproject.com/en/3.0/ref/models/fields/#django.db.models.ForeignKey.on\\_delete](https://docs.djangoproject.com/en/3.0/ref/models/fields/#django.db.models.ForeignKey.on_delete)

Instance	Due_back	Status
1		Available
2	next week	On Loan
3	who knows	On Loan

Title	ISBN
Wisdom of Crowds	385721706
Introduction to Networking	9781511654944
Introducción a las Redes	9781523627516

Language
en
es

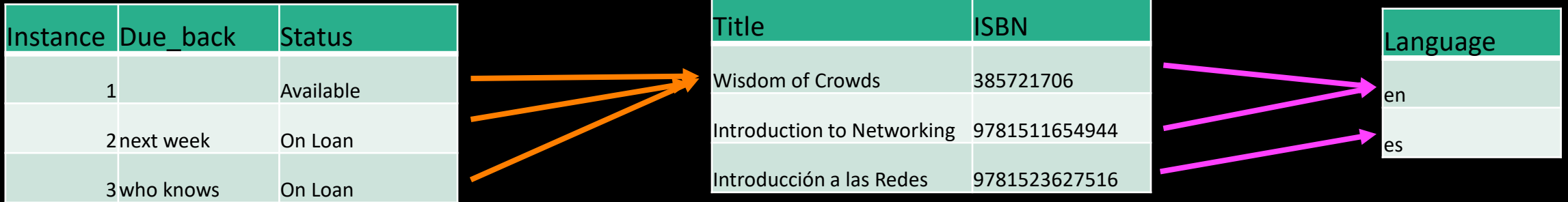
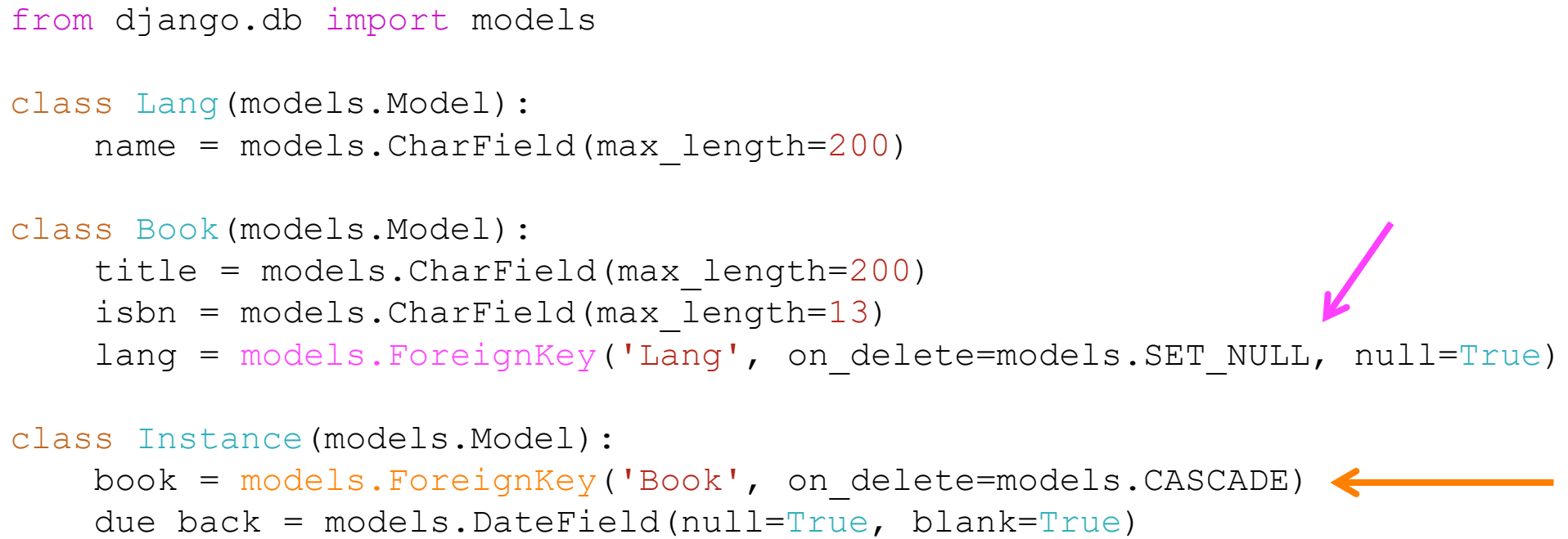
```

from django.db import models

class Lang(models.Model):
    name = models.CharField(max_length=200)

class Book(models.Model):
    title = models.CharField(max_length=200)
    isbn = models.CharField(max_length=13)
    lang = models.ForeignKey('Lang', on_delete=models.SET_NULL, null=True)

class Instance(models.Model):
    book = models.ForeignKey('Book', on_delete=models.CASCADE)
    due_back = models.DateField(null=True, blank=True)
  
```

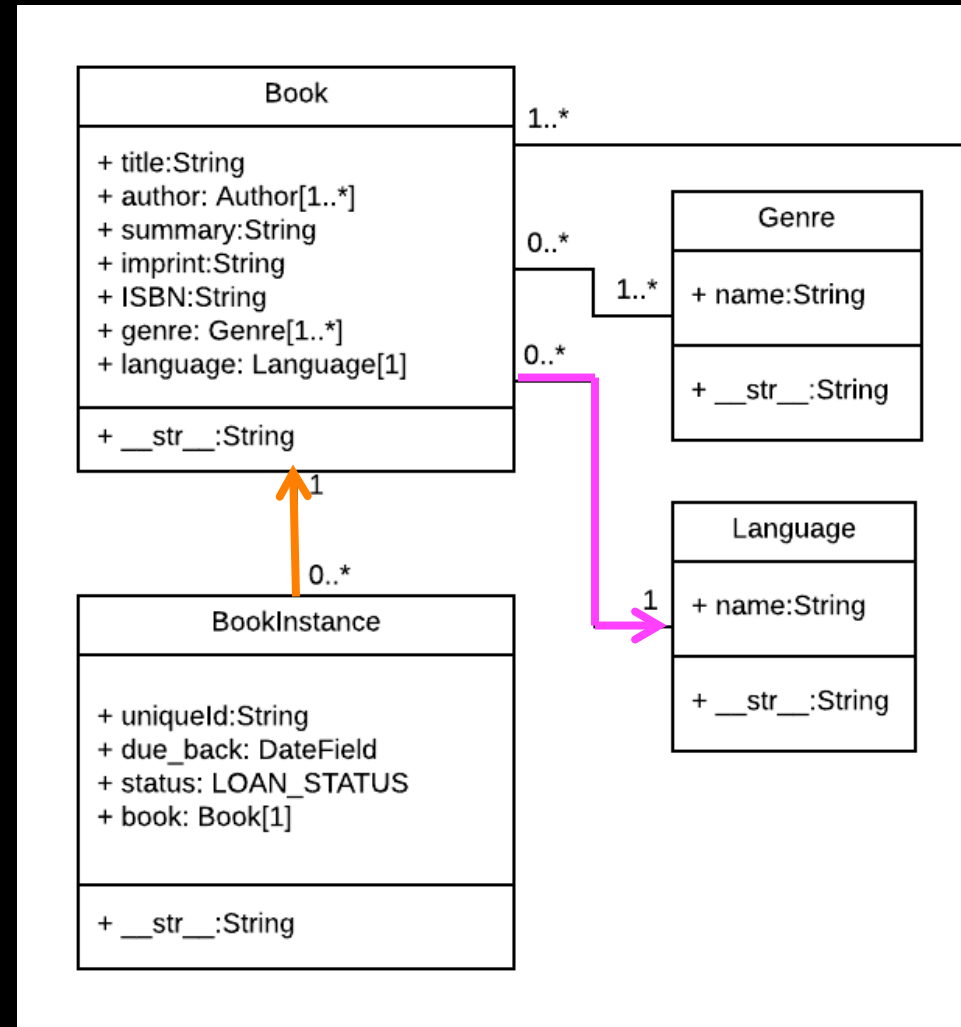
# Using Models in the Django Shell



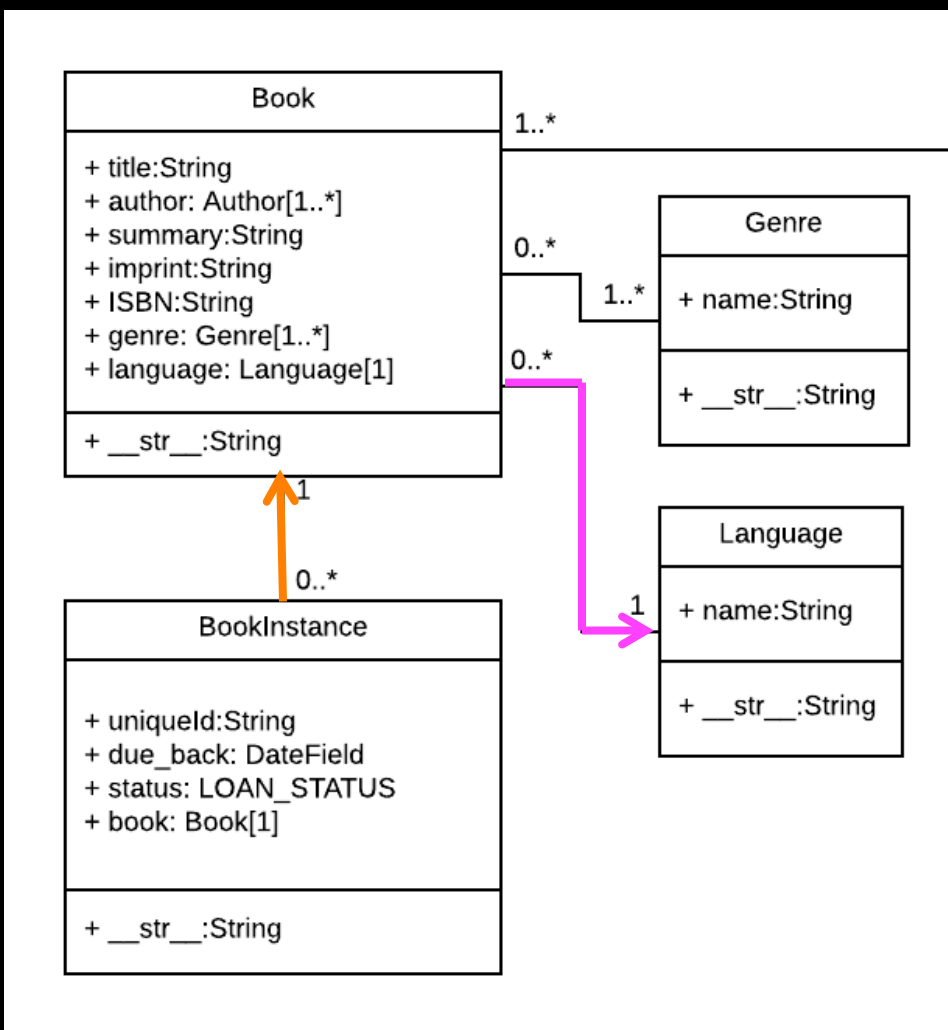
```

$ cd ~/dj4e-samples
$ python3 manage.py shell
>>> from bookone.models import Book, Lang, Instance
>>> z = Lang(name='en')
>>> z.save()
>>> z.id
1
>>> x = Book(title='PY4E', isbn='42', lang=z)
>>> x.save()
>>> x.id
1
>>> a = Instance(due_back="2020-07-06", book=x)
>>> a.save()
>>> a.id
1
>>> quit()
$

```



```
$ cd ~/dj4e-samples
$ python3 manage.py shell
>>> from bookone.models import Book, Lang, Instance
>>> x = Book.objects.get(pk=1)
>>> x
<Book: Book object (1)>
>>> x.title
'PY4E'
>>> x.lang.name
'en'
>>> y = Instance.objects.get(pk=1)
>>> y.due_back
datetime.date(2020, 7, 6)
>>> y.book.title
'PY4E'
>>> y.book.lang.name
'en'
>>> quit()
$
```



# Demo Batch Loading from CSV

<https://github.com/csev/dj4e-samples/tree/master/samples/scripts>

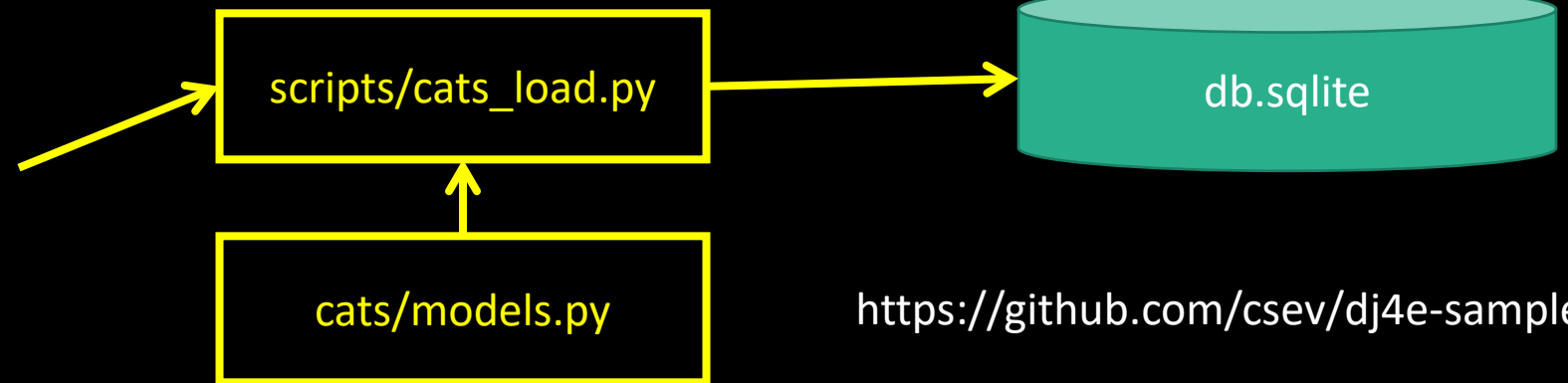
<https://django-extensions.readthedocs.io/en/latest/runscript.html>

# Loading Data From A File

- Sometimes we need to pre-load data into our Django database
- This data might come from an API or file
- We need to write a Python program to function like the Django shell

`cats/meow.csv`

```
Name,Breed,Weight  
Abby,Sphinx,6.4  
Annie,Burmese,7.6  
Ash,Manx,7.8  
Athena,Manx,8.9
```



<https://github.com/csev/dj4e-samples>

# Installing django-extensions

```
dj4e-samples$ pip3 install django-extensions
Requirement already satisfied: django-extensions in
/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages
Requirement already satisfied: six>=1.2 in
/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages
dj4e-samples$
```

Note that this is installed already in dj4e-samples but for a new project you will need to install it yourself and edit **settings.py**

<https://django-extensions.readthedocs.io/en/latest/runscript.html>

# Include Extensions in Project Settings

[dj4e-samples/settings.py](#)

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
  
    [ ... ]  
  
    # Extensions - see requirements.txt  
    'django_extensions',  
    'crispy_forms',  
  
    [ ... ]  
  
    'home.apps.HomeConfig',  
  
    # Sample Applications - don't copy  
    'hello.apps.HelloConfig',  
    'getpost.apps.GetpostConfig',  
    'users.apps.UsersConfig',  
  
    [ ... ]  
]
```

# Make a scripts folder

```
dj4e-samples$ mkdir scripts  
dj4e-samples$ touch scripts/__init__.py
```

We place empty `__init__.py` files in folders to indicate to Python that they contain files that hold modules and as such are suitable for importing into a Python application.

<http://effbot.org/pyfaq/what-is-init-py-used-for.htm>

# The Data File

```
dj4e-samples$ cat cats/meow.csv
```

```
Name,Breed,Weight
```

```
Abby,Sphinx,6.4
```

```
Annie,Burmese,7.6
```

```
Ash,Manx,7.8
```

```
Athena,Manx,8.9
```

```
dj4e-samples$
```

[https://en.wikipedia.org/wiki/Cat\\_\(Unix\)](https://en.wikipedia.org/wiki/Cat_(Unix))



## scripts/cats\_load.py

```
import csv
from cats.models import Cat, Breed

def run():
    fhand = open('cats/meow.csv')
    reader = csv.reader(fhand)
    next(reader) # Advance past the header

    Cat.objects.all().delete()
    Breed.objects.all().delete()

    # Name,Breed,Weight
    # Abby,Sphinx,6.4
    # Annie,Burmese,7.6
    # Ash,Manx,7.8

    for row in reader:
        print(row)

        b, created = Breed.objects.get_or_create(name=row[1])

        c = Cat(nickname=row[0], breed=b, weight=row[2])
        c.save()
```

```
from django.db import models

class Breed(models.Model):
    name = models.CharField(max_length=200)

class Cat(models.Model):
    nickname = models.CharField(max_length=200)
    breed = models.ForeignKey('Breed', on_delete=models.CASCADE, null=False)
    weight = models.FloatField()
```

```
dj4e-samples$ python3 manage.py runscript cats_load
```

```
['Abby', 'Sphinx', '6.4']
```

```
['Annie', 'Burmese', '7.6']
```

```
['Ash', 'Manx', '7.8']
```

```
['Athena', 'Manx', '8.9']
```

```
['Baby', 'Tabby', '6.9']
```

```
['Bagheera', 'Sphinx', '6.3']
```

```
dj4e-samples$
```

```
for row in reader:
```

```
    print(row)
```

```
    b, created = Breed.objects.get_or_create(name=row[1])
```

```
    c = Cat(nickname=row[0], breed=b, weight=row[2])
```

```
    c.save()
```

# Acknowledgements / Contributions

These slides are Copyright 2019- Charles R. Severance ([www.dr-chuck.com](http://www.dr-chuck.com)) as part of [www.dj4e.com](http://www.dj4e.com) and made available under a Creative Commons Attribution 4.0 License. Please maintain this last slide in all copies of the document to comply with the attribution requirements of the license. If you make a change, feel free to add your name and organization to the list of contributors on this page as you republish the materials.

Initial Development: Charles Severance, University of Michigan School of Information

Insert new Contributors and Translators here including names and dates

Continue new Contributors and Translators here