**icontains**

Case-insensitive containment test.

Example:

Entry.objects.get(headline\_\_icontains='Lennon')

SQL equivalent:

SELECT ... WHERE headline ILIKE '**%Le**nnon%';

**e.g:**

>>> from blogApp.models import Post

>>> posts = Post.objects.all()

>>> for p in posts:

... print(p.title)

...

Progressively engineer collaborative

Progressively streamline viral deliverables

Competently implement excellent

Compellingly formulate [Updated#2]

Continually pesontificate distindctive updated

Compellingly restore ethical ROIs

>>> res = Post.objects.filter(title\_\_icontains='Compelling')

>>> res

<QuerySet [<Post: Compellingly formulate [Updated#2]>, <Post: Compellingly restore ethical ROIs>]>

### Methods that return new QuerySets[¶](https://docs.djangoproject.com/en/2.1/ref/models/querysets/#methods-that-return-new-querysets)

exclude()

Returns a new **QuerySet** containing objects that do *not* match the given lookup parameters.

This example excludes all entries whose **pub\_date** is later than 2005-1-3 OR whose headline is “Hello”:

Entry.objects.exclude(pub\_date\_\_gt=datetime.date(2005, 1, 3)).exclude(headline='Hello')

In SQL terms, that evaluates to:

SELECT ...

WHERE NOT pub\_date > '2005-1-3'

AND NOT headline = 'Hello'

If you need to execute more complex queries (for example, queries with OR statements), you can use [Q objects](https://docs.djangoproject.com/en/2.1/ref/models/querysets/#django.db.models.Q).

Example: Show list of posts that doesn’t contain ‘compelllingly’ and ‘progressively’ in its title:

>>> posts = Post.objects.all()

>>> for post in posts:

... print(post.title)

...  
Progressively engineer collaborative

Progressively streamline viral deliverables

Competently implement excellent

Compellingly formulate [Updated#2]

Continually pesontificate distindctive updated

Assertively negotiate client-based

Proactively simplify world-class action items via premier ROI

Synergistically seize

Compellingly restore ethical ROIs

>>> res = Post.objects.exclude(Q(title\_\_icontains='Compellingly')| Q(title\_\_icontains='progressively'))

>>> for i in res:

... print(i)

...

Competently implement excellent

Continually pesontificate distindctive updated

Assertively negotiate client-based

Proactively simplify world-class action items via premier ROI

Synergistically seize