

## Queries: more advanced filtering

Flask-SQLAlchemy allows more complex queries and operations such as checking whether a column starts, or ends, with some string. One can also order retrieved queries by some criterion. There are many more possible queries, but here we cover only some of them.

For example, to retrieve e-mails that end with `edu` we do:

```
education = Reader.query.filter(Reader.email.endswith('edu')).all()
```

To retrieve all the readers with e-mails that contain a `.` before the `@` symbol we use `.like()`:

```
emails = Reader.query.filter(Reader.email.like('%.%@%')).all()
```

You might recognize the `like` operator from SQL. It is used to search for a specified pattern in a column. The wildcard `%` represents zero, one, or multiple characters.

In the two examples above, we used methods on the column of the table (SQLAlchemy's `ColumnElement`).

To order books by year we use the `.order_by()` method on `Query`:

```
ordered_books = Book.query.order_by(Book.year).all()
```

We suggest checking the SQLAlchemy Core + ORM documentation to see other querying options.

## Instructions

1.

Using `filter` and column-based criterion `endswith`, write a query that retrieves all the readers with the surname that ends with an `'s'` and assign the result to a variable called `s_names`.

Stuck? Get a hint

2.

Using `like`, write a query that retrieves all the readers with e-mails having "sample" instead of "example" just after the "@" symbol, and assign the result to a variable called `sample_emails`.

Stuck? Get a hint

3.

Using `order_by`, write a query that orders the reviews by the number of stars they have in ascending order, and assign the result to a variable called `ordered_reviews`.

```
from app import db, Book, Reader, Review
```

```
#retrieve all reader with .edu e-mails
education = Reader.query.filter(Reader.email.endswith('edu')).all()
print(education)

#retrieve all readers with e-mails that contain a . before the @ symbol
emails = Reader.query.filter(Reader.email.like('%.%@%')).all()
print("\nReaders with e-mails having a . before the @ symbol:")
for e in emails:
    print(e.email)

#order all books by year
ordered_books = Book.query.order_by(Book.year).all()
print("\nBooks ordered by year:")
for book in ordered_books:
    print(book.title, book.year)

print("\nCheckpoint 1: your code here below:")
s_names = Reader.query.filter(Reader.surname.endswith('s')).all()
#print(s_names)

print("\nCheckpoint 2: your code here below:")
sample_emails = Reader.query.filter(Reader.email.like('%@sample%')).all()
print(sample_emails)

print("\nCheckpoint 3: your code here below:")
ordered_reviews = Review.query.order_by(Review.stars).all()
print(ordered_reviews)
```