Session: Removing database entries

Removing entries is an important aspect of database management and is used often in real-world applications. Users unsubscribe from services, products are removed from web applications, and some relationships are lost (unfollowing other users).

However, before we proceed, we need to be careful about one-to-many relationships. If we remove a reader, we would expect that all the reader's reviews are also removed from our database. Similarly, removing a book should also remove all the reviews for that book. This procedure is called cascading deletion. Unfortunately, the way we previously declared our `Reader` and `Book` models will not perform the cascading deletion by default. To enable cascading deletions, we did a naive solution in this exercise by changing our models and re-initializing the database. In practice, database migration management is used to update a database schema.

To enable cascade deletions, we changed the models in the **app.py** by adding the `cascade` parameter to the `.relationship()` fields of `Reader` and `Book` models:

```
reviews = db.relationship('Review', backref='reviewer', lazy='dynamic',
cascade = 'all, delete, delete-orphan')
```

In contrast, removing a review does not have any other cascading consequences on `Book` and `Reader` tables. Hence, specifying the cascading deletion option in `Review` is not needed.

Finally, to remove a reader with `id = 753` we use the following command:

```
db.session.delete(Reader.query.get(753))
```

When you run **playground.py** you see that we print all the readers, all the reviews before and after the deletion. You can notice that when the reader with `id = 753` is deleted, all their reviews are deleted as well. Refer to the image on the right to see the initial entries of some database tables.

Instructions

1.

Use the `delete()` command in combination with `get()` to delete a reader entry with `id = 123` from the database in the image on the right.

```
reviews = db.relationship('Review', backref='reviewer', lazy = 'dynamic',
cascade = "all, delete, delete-orphan")
```

```
from app import db, Book, Reader, Review #notice we import db here as well

#let us first print all the readers current in the database (image on the
right)
for reader in Reader.query.all():
   print(reader)

#print all the reviews
print("\nAll the current reviews:")
for review in Review.query.all():
  print(review)

#delete reader with id = 753 (Nova Yeni, nova.yeni@example.com)
db.session.delete(Reader.query.get(753))

#print readers again to validate that the reader is indeed deleted
#Checkpoint 1:
print("\nReaders after deleting a rader with id = 123")
for reader in Reader.query.all():
   print(reader)

#print reviews to see that all the reviews made by reader id = 123 are
deleted
#(see the image on the right)
db.session.delete(Reader.query.get(123))
#print all the reviews
print("\nAll the current reviews:")
for review in Review.query.all():
  print(review)
```