Session: updating existing entries

Sometimes you will need to update a certain column value of an entry in your database. This is rather easy in the context of SQLAlchemy ORM and is done in the same way you would change Python object's attribute.

The commands below change the email of a reader with `id=3` and commit the changes to the database:

```
reader = Reader.query.get(3)
reader.email = “new_email@example.com”
db.session.commit()
```

If you want to undo the update, you can use

```
db.session.rollback()
```

instead of committing.

Instructions

1.

Using the `get` method, fetch a book entry with `id = 19` and assign it to the variable called `book_19`.

Stuck? Get a hint

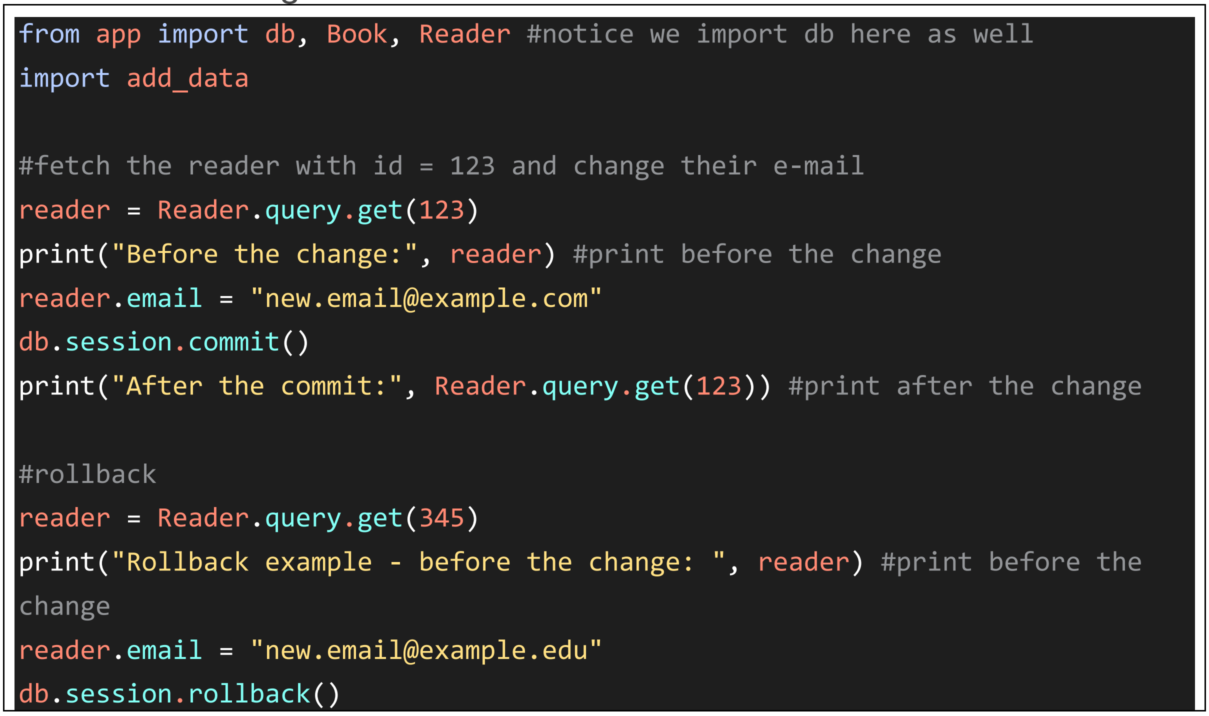
2.

Change the `month` attribute of `book_19` to `June`.

Stuck? Get a hint

3.

Commit the change.



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

#fetch the reader with id = 123 and change their e-mail
reader = Reader.query.get(123)
print("Before the change:", reader) #print before the change
reader.email = "new.email@example.com"
db.session.commit()
print("After the commit:", Reader.query.get(123)) #print after the change

#rollback
reader = Reader.query.get(345)
print("Rollback example - before the change: ", reader) #print before the
change
reader.email = "new.email@example.edu"
db.session.rollback()
```

```
print("Rollback example - after the rollback: ", Reader.query.get(345))
#print after the change

print("\nCheckpoint 1: use get to fetch a book entry:")
book_19 = Book.query.get(19)

print("\nCheckpoint 2: modify the month attribute to June:")
book_19.month = "June"

print("\nCommit the change:")
db.session.commit()
```