

DUT STID, Université de la Côte d'Azur

Bases de données avancées

ORM

Prof. Dario Malchiodi



UNIVERSITÀ DEGLI STUDI DI MILANO
DIPARTIMENTO DI INFORMATICA

UNIVERSITÉ
CÔTE D'AZUR 

SQLAlchemy

In [121]:

```
import os
import pandas as pd
from sqlalchemy import create_engine

db_name = 'superheroes'

connection_uri = ('mysql+mysqlconnector://' +
                  os.environ['MYSQL_USER'] + ':' +
                  os.environ['MYSQL_PASSWORD'] + '@' +
                  os.environ['MYSQL_HOST_IP'] + ':' +
                  os.environ['MYSQL_PORT'] + '/' + db_name)
```

In [123]:

```
engine = create_engine(connection_uri, echo=False)
```

In [129]:

```
pd.read_sql('SELECT * FROM heroes where eye_color="Red";', engine, index_col='id')
```

Out[129]:

id	name	identity	birth_place	publisher	height	weight	gender	first_appearanc
21	Zoom	Hunter Zolomon	None	DC Comics	185.90	81.93	M	NaI
43	Wonder Man	Simon Williams	Paterson, New Jersey	Marvel Comics	188.90	171.57	M	NaI
45	Warlock	Adam Warlock	The Beehive, Shard Island, Atlantic Ocean	Marvel Comics	188.22	108.14	M	NaI
53	Vision II	Vision	Place of birth unknown	Marvel Comics	191.39	135.51	None	NaI
71	Ultron	Ultron	Created in Cresskill, New Jersey	Marvel Comics	206.90	331.61	M	1968.
92	Thanos	Thanos	Titan	Marvel Comics	201.26	443.19	M	NaI
99	T-800	Cyberdyne Systems Series 800 Terminator Model 101	None	Dark Horse Comics	NaN	176.28	M	1984.
100	T-850	Cyberdyne Systems Series 850 Terminator	None	Dark Horse Comics	NaN	198.72	M	2003.
104	Swamp Thing	Alec Holland	None	DC Comics	NaN	NaN	M	1972.
115	Steppenwolf	None	None	DC Comics	183.36	91.04	M	NaI
126	Spider-Woman IV	Charlotte Witter	None	Marvel Comics	178.09	58.78	F	NaI
160	Shadow King	Amahl Farouk	None	Marvel Comics	185.48	149.19	None	1979.
173	Sasquatch	Walter Langkowski	Edmonton, Alberta, Canada	Marvel Comics	305.02	900.70	M	NaI
246	Onslaught	Onslaught	None	Marvel Comics	305.41	405.08	M	1996.
249	Omega Red	Arkady Gregorivich	None	Marvel Comics	211.79	191.40	M	NaI
288	Mister Sinister	Nathaniel Essex	London, England	Marvel Comics	196.89	128.37	M	1987.
295	Miss Martian	M'gann M'orzz	Mars	DC Comics	178.49	61.25	F	2006.
316	Martian Manhunter	J'onnn J'onzz	None	DC Comics	201.86	135.18	M	1955.

id	name	identity	birth_place	publisher	height	weight	gender	first_appearanc
322	Machine Man	X-51, Aaron Stack	Central City, California	Marvel Comics	183.42	383.81	None	NaI
323	Man-Thing	Dr. Theodore Sallis	None	Marvel Comics	213.38	225.60	M	1971.
335	Lobo	Lobo	None	DC Comics	229.40	288.82	M	NaI
352	Klaw	Ulysses Klaw	Vlaardingen, Netherlands	Marvel Comics	188.61	97.99	M	1966.
355	Lizard	Curtis Connors	Coral Gables, Florida	Marvel Comics	203.53	230.60	M	NaI
359	Kilowog	None	Bolovax Vik	DC Comics	234.39	324.10	M	1986.
360	Killer Croc	Waylon Jones	None	DC Comics	244.58	356.55	M	1983.
374	Jubilee	Jubilation Lee	Beverly Hills, California	Marvel Comics	165.92	52.34	F	1989.
430	Hellstorm	Daimon Hellstrom	Greentown, Massachusetts	Marvel Comics	185.02	81.45	M	NaI
438	Hawk	Henry Hall	None	DC Comics	185.91	89.40	M	1968.
469	Gambit	Remy Etienne LeBeau	New Orleans, Louisiana	Marvel Comics	185.55	81.57	M	2006.
470	Ghost Rider	Johnny Blaze	Waukegan, Illinois	Marvel Comics	188.51	99.20	M	1972.
484	Fixer	Paul Norbert Ebersol	Dayton, Ohio	Marvel Comics	NaN	NaN	None	NaI
488	Fin Fang Foom	None	None	Marvel Comics	975.37	19.00	M	NaI
499	Evilhawk	Dargin Bokk	None	Marvel Comics	191.25	106.78	M	1992.
502	Etrigan	Jason Blood	Hell	DC Comics	193.76	203.50	M	1972.
511	Drax the Destroyer	Arthur Sampson Douglas	Burbank, California	Marvel Comics	193.01	306.42	M	1973.
513	Doomsday	Doomsday	Krypton	DC Comics	244.01	412.67	M	1992.
526	Demogoblin	None	None	Marvel Comics	185.05	95.29	M	1992.
540	Darkseid	Uxas	None	DC Comics	267.37	817.57	M	1970.
556	Copycat	Vanessa Carlysle	None	Marvel Comics	183.47	67.55	F	1991.
581	Captain Planet	None	None	Marvel Comics	NaN	NaN	M	NaI
628	Blackwulf	Lucian	Armechadon	Marvel Comics	188.17	88.64	M	1994.

id	name	identity	birth_place	publisher	height	weight	gender	first_appearanc
631	Blackout	None	None	Marvel Comics	191.20	104.09	M	1990.
644	Black Abbott	None	None	Marvel Comics	NaN	NaN	M	1984.
700	Apocalypse	En Sabah Nur	Akkaba, Egypt	Marvel Comics	213.77	135.62	M	1986.
715	Amazo	None	None	DC Comics	257.49	173.95	M	1960.

In [1]:

```
import sqlalchemy
sqlalchemy.__version__
```

Out[1]:

'1.2.7'

In [65]:

```
engine = create_engine('mysql+mysqlconnector://orm:Passw0rd.@localhost:3306/ORM',
                        echo=False)
```

In [66]:

```
from sqlalchemy.ext.declarative import declarative_base
Base = declarative_base()
```

In [67]:

```
from sqlalchemy import Column, Integer, String
class User(Base):
    __tablename__ = 'users'

    id = Column(Integer, primary_key=True)
    name = Column(String(100))
    fullname = Column(String(250))
    nickname = Column(String(100))

    def __repr__(self):
        return "<User(name='{}', fullname='{}', nickname='{}')>".format(
            self.name, self.fullname, self.nickname)
```

In [68]:

```
User.__table__
```

Out[68]:

```
Table('users', MetaData(bind=None), Column('id', Integer(), table=<users>, primary_key=True, nullable=False), Column('name', String(length=100), table=<users>), Column('fullname', String(length=250), table=<users>), Column('nickname', String(length=100), table=<users>), schema=None)
```

In [6]:

```
Base.metadata.create_all(engine)
```

In [7]:

```
ed_user = User(name='ed', fullname='Ed Jones', nickname='edsnickname')
```

In [8]:

```
(ed_user.name, ed_user.nickname)
```

Out[8]:

```
('ed', 'edsnickname')
```

In [9]:

```
str(ed_user.id)
```

Out[9]:

```
'None'
```

In [10]:

```
from sqlalchemy.orm import sessionmaker
Session = sessionmaker(bind=engine)
session = Session()
```

In [11]:

```
session.add(ed_user)
```

In [12]:

```
our_user = session.query(User).filter_by(name='ed').first()
our_user
```

Out[12]:

```
<User(name='ed', fullname='Ed Jones', nickname='edsnickname')>
```

In [13]:

```
ed_user is our_user
```

Out[13]:

True

In [14]:

```
session.add_all([
    User(name='wendy', fullname='Wendy Williams', nickname='windy'),
    User(name='mary', fullname='Mary Contrary', nickname='mary'),
    User(name='fred', fullname='Fred Flintstone', nickname='freddy')])
```

In [15]:

```
ed_user.nickname = 'eddie'
```

In [16]:

```
session.dirty
```

Out[16]:

```
IdentitySet([<User(name='ed', fullname='Ed Jones', nickname='eddie')>])
```

In [17]:

```
session.new
```

Out[17]:

```
IdentitySet([<User(name='wendy', fullname='Wendy Williams', nickname='windy')>, <User(name='mary', fullname='Mary Contrary', nickname='mary')>, <User(name='fred', fullname='Fred Flintstone', nickname='freddy')>])
```

In [18]:

```
session.commit()
```

In [19]:

```
ed_user.id
```

Out[19]:

28

In [20]:

```
ed_user.name = 'Edwardo'  
fake_user = User(name='fakeuser', fullname='Invalid', nickname='12345')  
session.add(fake_user)
```

In [21]:

```
session.query(User).filter(User.name.in_(['Edwardo', 'fakeuser'])).all()
```

Out[21]:

```
[<User(name='Edwardo', fullname='Ed Jones', nickname='eddie')>,  
<User(name='fakeuser', fullname='Invalid', nickname='12345')>]
```

In [22]:

```
session.rollback()
```

In [23]:

```
ed_user.name
```

Out[23]:

```
'ed'
```

In [24]:

```
fake_user in session
```

Out[24]:

```
False
```

In [25]:

```
session.query(User).filter(User.name.in_(['ed', 'fakeuser'])).all()
```

Out[25]:

```
[<User(name='ed', fullname='Ed Jones', nickname='eddie')>]
```

In [26]:

```
for instance in session.query(User).order_by(User.id):  
    print(instance.name, instance.fullname)
```

```
ed Ed Jones  
wendy Wendy Williams  
mary Mary Contrary  
fred Fred Flintstone
```

In [27]:

```
for name, fullname in session.query(User.name, User.fullname):
    print(name, fullname)
```

```
ed Ed Jones
wendy Wendy Williams
mary Mary Contrary
fred Fred Flintstone
```

In [28]:

```
for row in session.query(User, User.name).all():
    print(row.User, row.name)
```

```
<User(name='ed', fullname='Ed Jones', nickname='eddie')> ed
<User(name='wendy', fullname='Wendy Williams', nickname='windy')> wend
y
<User(name='mary', fullname='Mary Contrary', nickname='mary')> mary
<User(name='fred', fullname='Fred Flintstone', nickname='freddy')> fre
d
```

In [29]:

```
for u in session.query(User).order_by(User.id)[1:3]:
    print(u)
```

```
<User(name='wendy', fullname='Wendy Williams', nickname='windy')>
<User(name='mary', fullname='Mary Contrary', nickname='mary')>
```

In [30]:

```
for name, in session.query(User.name).filter_by(fullname='Ed Jones'):
    print(name)
```

```
ed
```

In [31]:

```
for user in session.query(User).\
    filter(User.name=='ed').\
    filter(User.fullname=='Ed Jones'):
    print(user)
```

```
<User(name='ed', fullname='Ed Jones', nickname='eddie')>
```

In [32]:

```
for user in session.query(User).filter(User.name.like('%ed%')):
    print(user)
```

```
<User(name='ed', fullname='Ed Jones', nickname='eddie')>
<User(name='fred', fullname='Fred Flintstone', nickname='freddy')>
```


In [33]:

```
for user in session.query(User).filter(User.name == None):
    print(user)
```

In [34]:

```
query = session.query(User).filter(User.name.like('%ed')).order_by(User.id)
query.all()
```

Out[34]:

```
[<User(name='ed', fullname='Ed Jones', nickname='eddie')>,
 <User(name='fred', fullname='Fred Flintstone', nickname='freddy')>]
```

In [35]:

```
query.first()
```

Out[35]:

```
<User(name='ed', fullname='Ed Jones', nickname='eddie')>
```

In [36]:

```
session.query(User).filter(User.name.like('%ed')).count()
```

Out[36]:

```
2
```

In [37]:

```
from sqlalchemy import ForeignKey
from sqlalchemy.orm import relationship

class Address(Base):
    __tablename__ = 'addresses'
    id = Column(Integer, primary_key=True)
    email_address = Column(String(200), nullable=False)
    user_id = Column(Integer, ForeignKey('users.id'))

    user = relationship("User", back_populates="addresses")

    def __repr__(self):
        return "<Address(email_address='{ }')>".format(self.email_address)

User.addresses = relationship(
    "Address", order_by=Address.id, back_populates="user")
```

In [38]:

```
Base.metadata.create_all(engine)
```

In [39]:

```
jack = User(name='jack', fullname='Jack Bean', nickname='gjffdd')
jack.addresses
```

Out[39]:

```
[]
```

In [40]:

```
jack.addresses = [Address(email_address='jack@google.com'),
                  Address(email_address='j25@yahoo.com')]
```

In [41]:

```
session.add(jack)
session.commit()
```

In [42]:

```
jack = session.query(User).filter_by(name='jack').one()
jack
```

Out[42]:

```
<User(name='jack', fullname='Jack Bean', nickname='gjffdd')>
```

In [43]:

```
jack.addresses
```

Out[43]:

```
[<Address(email_address='jack@google.com')>,
 <Address(email_address='j25@yahoo.com')>]
```

In [44]:

```
(session.query(User).join(Address)
 .filter(Address.email_address=='jack@google.com')
 .all())
```

Out[44]:

```
[<User(name='jack', fullname='Jack Bean', nickname='gjffdd')>]
```

In []: