

SQLAlchemy/database.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # database.py
5: # Author: Bob Dondero
6: #-----
7:
8: import sqlalchemy
9: import sqlalchemy.orm
10:
11: class Base(sqlalchemy.orm.DeclarativeBase):
12:     pass
13:
14: class Book (Base):
15:     __tablename__ = 'books'
16:     isbn = sqlalchemy.Column(sqlalchemy.String,
17:                             primary_key=True, nullable=False)
18:     title = sqlalchemy.Column(sqlalchemy.String)
19:     quantity = sqlalchemy.Column(sqlalchemy.Integer)
20:
21: class Author (Base):
22:     __tablename__ = 'authors'
23:     isbn = sqlalchemy.Column(sqlalchemy.String,
24:                             sqlalchemy.ForeignKey('books.isbn'),
25:                             primary_key=True, nullable=False)
26:     author = sqlalchemy.Column(sqlalchemy.String,
27:                               primary_key=True, nullable=False)
28:     #books = sqlalchemy.orm.relationship('Book', backref='authors')
29:
30: class Zipcode (Base):
31:     __tablename__ = 'zipcodes'
32:     zipcode = sqlalchemy.Column(sqlalchemy.String,
33:                                primary_key=True, nullable=False)
34:     city = sqlalchemy.Column(sqlalchemy.String)
35:     state = sqlalchemy.Column(sqlalchemy.String)
36:
37: class Customer (Base):
38:     __tablename__ = 'customers'
39:     custid = sqlalchemy.Column(sqlalchemy.String,
40:                                primary_key=True, nullable=False)
41:     custname = sqlalchemy.Column(sqlalchemy.String)
42:     street = sqlalchemy.Column(sqlalchemy.String)
43:     zipcode = sqlalchemy.Column(sqlalchemy.String,
44:                                sqlalchemy.ForeignKey('zipcodes.zipcode'))
45:     #zipcodes=sqlalchemy.orm.relationship('Zipcode',backref='customers')
46:
47: class Order (Base):
48:     __tablename__ = 'orders'
49:     isbn = sqlalchemy.Column(sqlalchemy.String,
50:                             sqlalchemy.ForeignKey('books.isbn'),
51:                             primary_key=True, nullable=False)
52:     custid = sqlalchemy.Column(sqlalchemy.String,
53:                               sqlalchemy.ForeignKey('customers.custid'),
54:                               primary_key=True, nullable=False)
55:     quantity = sqlalchemy.Column(sqlalchemy.Integer)

```

blank (Page 1 of 1)

1: This page is intentionally blank.

SQLAlchemy/create.py (Page 1 of 2)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # create.py
5: # Author: Bob Dondero
6: #-----
7:
8: import os
9: import sys
10: import sqlalchemy
11: import sqlalchemy.orm
12: import dotenv
13: import database
14:
15: dotenv.load_dotenv()
16: DATABASE_URL = os.getenv('DATABASE_URL', 'sqlite:///bookstore.sqlite')
17: DATABASE_URL = DATABASE_URL.replace('postgres://', 'postgresql://')
18: _DATABASE_URL = _DATABASE_URL.replace(
19:     'postgresql://', 'postgresql+psycopg://')
20:
21: engine = sqlalchemy.create_engine(DATABASE_URL)
22:
23: #-----
24:
25: def main():
26:
27:     if len(sys.argv) != 1:
28:         print(f'usage: python {sys.argv[0]}', file=sys.stderr)
29:         sys.exit(1)
30:
31:     try:
32:
33:         database.Base.metadata.drop_all(engine)
34:         database.Base.metadata.create_all(engine)
35:
36:         with sqlalchemy.orm.Session(engine) as session:
37:
38:             #-----
39:
40:             book = database.Book(isbn=123,
41:                                 title='The Practice of Programming', quantity=500)
42:             session.add(book)
43:             book = database.Book(isbn=234,
44:                                 title='The C Programming Language', quantity=800)
45:             session.add(book)
46:             book = database.Book(isbn=345,
47:                                 title='Algorithms in C', quantity=650)
48:             session.add(book)
49:             session.commit()
50:
51:             #-----
52:
53:             author = database.Author(isbn=123, author='Kernighan')
54:             session.add(author)
55:             author = database.Author(isbn=123, author='Pike')
56:             session.add(author)
57:             author = database.Author(isbn=234, author='Kernighan')
58:             session.add(author)
59:             author = database.Author(isbn=234, author='Ritchie')
60:             session.add(author)
61:             author = database.Author(isbn=345, author='Sedgewick')
62:             session.add(author)
63:             session.commit()
64:
65:             #-----

```

SQLAlchemy/create.py (Page 2 of 2)

```

66:
67:         zipcode = database.Zipcode(zipcode='08540',
68:                                     city='Princeton', state='NJ')
69:         session.add(zipcode)
70:         zipcode = database.Zipcode(zipcode='02138',
71:                                     city='Cambridge', state='MA')
72:         session.add(zipcode)
73:         zipcode = database.Zipcode(zipcode='02142',
74:                                     city='Cambridge', state='MA')
75:         session.add(zipcode)
76:         session.commit()
77:
78:         #-----
79:
80:         customer = database.Customer(custid='111',
81:                                     custname='Princeton', street='114 Nassau St',
82:                                     zipcode='08540')
83:         session.add(customer)
84:         customer = database.Customer(custid='222',
85:                                     custname='Harvard', street='1256 Mass Ave',
86:                                     zipcode='02138')
87:         session.add(customer)
88:         customer = database.Customer(custid='333',
89:                                     custname='MIT', street='292 Main St',
90:                                     zipcode='02142')
91:         session.add(customer)
92:         session.commit()
93:
94:         #-----
95:
96:         order = database.Order(isbn='123', custid='222',
97:                                quantity=20)
98:         session.add(order)
99:         order = database.Order(isbn='345', custid='222',
100:                                quantity=100)
101:         session.add(order)
102:         order = database.Order(isbn='123', custid='111',
103:                                quantity=30)
104:         session.add(order)
105:         session.commit()
106:
107:     except Exception as ex:
108:         print(f'{sys.argv[0]}: {ex}', file=sys.stderr)
109:         sys.exit(1)
110:
111: #-----
112:
113: if __name__ == '__main__':
114:     main()

```

SQLAlchemy/display.py (Page 1 of 2)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # display.py
5: # Author: Bob Dondero
6: #-----
7:
8: import os
9: import sys
10: import sqlalchemy
11: import sqlalchemy.orm
12: import dotenv
13: import database
14:
15: dotenv.load_dotenv()
16: DATABASE_URL = os.getenv('DATABASE_URL', 'sqlite:///bookstore.sqlite')
17: DATABASE_URL = DATABASE_URL.replace('postgres://', 'postgresql://')
18: _DATABASE_URL = _DATABASE_URL.replace(
19:     'postgresql://', 'postgresql+psycopg://')
20:
21: engine = sqlalchemy.create_engine(DATABASE_URL)
22:
23: #-----
24:
25: def main():
26:
27:     if len(sys.argv) != 1:
28:         print(f'usage: python {sys.argv[0]}', file=sys.stderr)
29:         sys.exit(1)
30:
31:     try:
32:
33:         with sqlalchemy.orm.Session(engine) as session:
34:
35:             print('-----')
36:             print('books')
37:             print('-----')
38:             query = session.query(database.Book)
39:             table = query.all()
40:             for row in table:
41:                 print(row.isbn, row.title, row.quantity)
42:
43:             print('-----')
44:             print('authors')
45:             print('-----')
46:             query = session.query(database.Author)
47:             table = query.all()
48:             for row in table:
49:                 print(row.isbn, row.author)
50:
51:             print('-----')
52:             print('zipcodes')
53:             print('-----')
54:             query = session.query(database.Zipcode)
55:             table = query.all()
56:             for row in table:
57:                 print(row.zipcode, row.city, row.state)
58:
59:             print('-----')
60:             print('customers')
61:             print('-----')
62:             query = session.query(database.Customer)
63:             table = query.all()
64:             for row in table:
65:                 print(row.custid, row.custname, row.street, row.zipcode)

```

SQLAlchemy/display.py (Page 2 of 2)

```

66:
67:             print('-----')
68:             print('orders')
69:             print('-----')
70:             query = session.query(database.Order)
71:             table = query.all()
72:             for row in table:
73:                 print(row.isbn, row.custid, row.quantity)
74:
75:             engine.dispose()
76:
77:     except Exception as ex:
78:         print(f'{sys.argv[0]}: {ex}', file=sys.stderr)
79:         sys.exit(1)
80:
81: #-----
82:
83: if __name__ == '__main__':
84:     main()

```

SQLAlchemy/authorsearch.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # authorsearch.py
5: # Author: Bob Dondero
6: #-----
7:
8: import os
9: import sys
10: import sqlalchemy
11: import sqlalchemy.orm
12: import dotenv
13: import database
14:
15: dotenv.load_dotenv()
16: DATABASE_URL = os.getenv('DATABASE_URL', 'sqlite:///bookstore.sqlite')
17: DATABASE_URL = DATABASE_URL.replace('postgres://', 'postgresql://')
18: _DATABASE_URL = _DATABASE_URL.replace(
19:     'postgres://', 'postgresql+psycpg://')
20:
21: engine = sqlalchemy.create_engine(DATABASE_URL)
22:
23: #-----
24:
25: def main():
26:
27:     if len(sys.argv) != 2:
28:         print(f'usage: python {sys.argv[0]} author', file=sys.stderr)
29:         sys.exit(1)
30:
31:     author = sys.argv[1]
32:
33:     try:
34:         with sqlalchemy.orm.Session(engine) as session:
35:
36:             query = (session.query(database.Book.isbn,
37:                                     database.Book.title, database.Book.quantity)
38:                      .filter(database.Book.isbn == database.Author.isbn)
39:                      .filter(database.Author.author == author))
40:             print(query)
41:
42:             table = query.all()
43:             for row in table:
44:                 print('ISBN:', row.isbn)
45:                 print('Title:', row.title)
46:                 print('Quantity:', row.quantity)
47:                 print()
48:
49:     except Exception as ex:
50:         print(f'{sys.argv[0]}: {ex}', file=sys.stderr)
51:         sys.exit(1)
52:
53: #-----
54:
55: if __name__ == '__main__':
56:     main()

```

SQLAlchemy/order.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # order.py
5: # Author: Bob Dondero
6: #-----
7:
8: import os
9: import sys
10: import sqlalchemy
11: import sqlalchemy.orm
12: import dotenv
13: import database
14:
15: dotenv.load_dotenv()
16: DATABASE_URL = os.getenv('DATABASE_URL', 'sqlite:///bookstore.sqlite')
17: DATABASE_URL = DATABASE_URL.replace('postgres://', 'postgresql://')
18: _DATABASE_URL = _DATABASE_URL.replace(
19:     'postgres://', 'postgresql+psycpg://')
20:
21: engine = sqlalchemy.create_engine(DATABASE_URL)
22:
23: #-----
24:
25: def main():
26:
27:     if len(sys.argv) != 3:
28:         print(f'usage: python {sys.argv[0]} isbn custid',
29:               file=sys.stderr)
30:         sys.exit(1)
31:
32:     isbn = sys.argv[1]
33:     custid = sys.argv[2]
34:
35:     try:
36:         with sqlalchemy.orm.Session(engine) as session:
37:
38:             query = (session.query(database.Order)
39:                      .filter(database.Order.isbn == isbn)
40:                      .filter(database.Order.custid == custid))
41:
42:             row = query.one()
43:             row.quantity += 1
44:
45:             session.commit()
46:
47:     except Exception as ex:
48:         print(f'{sys.argv[0]}: {ex}', file=sys.stderr)
49:         sys.exit(1)
50:
51: #-----
52:
53: if __name__ == '__main__':
54:     main()

```

SQLAlchemy/purchase.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # purchase.py
5: # Author: Bob Dondero
6: #-----
7:
8: import os
9: import sys
10: import sqlalchemy
11: import sqlalchemy.orm
12: import dotenv
13: import database
14:
15: dotenv.load_dotenv()
16: DATABASE_URL = os.getenv('DATABASE_URL', 'sqlite:///bookstore.sqlite')
17: DATABASE_URL = DATABASE_URL.replace('postgres://', 'postgresql://')
18: _DATABASE_URL = _DATABASE_URL.replace(
19:     'postgresql://', 'postgresql+psycopg://')
20:
21: engine = sqlalchemy.create_engine(DATABASE_URL)
22:
23: #-----
24:
25: def main():
26:
27:     if len(sys.argv) != 3:
28:         print(f'usage: python {sys.argv[0]} isbn custid',
29:             file=sys.stderr)
30:         sys.exit(1)
31:
32:     isbn = sys.argv[1]
33:     custid = sys.argv[2]
34:
35:     try:
36:         with sqlalchemy.orm.Session(engine) as session:
37:
38:             query = (session.query(database.Order)
39:                 .filter(database.Order.isbn == isbn)
40:                 .filter(database.Order.custid == custid))
41:             row = query.one()
42:             row.quantity += 1
43:
44:             query = (session.query(database.Book)
45:                 .filter(database.Book.isbn == isbn))
46:             row = query.one()
47:             row.quantity -= 1
48:
49:             session.commit()
50:             print('Transaction committed.')
51:
52:     except Exception as ex:
53:         print(f'{sys.argv[0]}: {ex}', file=sys.stderr)
54:         sys.exit(1)
55:
56: #-----
57:
58: if __name__ == '__main__':
59:     main()

```

SQLAlchemy/recovery.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # recovery.py
5: # Author: Bob Dondero
6: #-----
7:
8: import os
9: import sys
10: import random
11: import sqlalchemy
12: import sqlalchemy.orm
13: import dotenv
14: import database
15:
16: dotenv.load_dotenv()
17: DATABASE_URL = os.getenv('DATABASE_URL', 'sqlite:///bookstore.sqlite')
18: DATABASE_URL = DATABASE_URL.replace('postgres://', 'postgresql://')
19: _DATABASE_URL = _DATABASE_URL.replace(
20:     'postgresql://', 'postgresql+psycopg://')
21:
22: engine = sqlalchemy.create_engine(DATABASE_URL)
23:
24: #-----
25:
26: def main():
27:
28:     if len(sys.argv) != 1:
29:         print(f'usage: python {sys.argv[0]}', file=sys.stderr)
30:         sys.exit(1)
31:
32:     try:
33:         with sqlalchemy.orm.Session(engine) as session:
34:
35:             for i in range(20):
36:
37:                 query = (session.query(database.Order)
38:                     .filter(database.Order.isbn == '123')
39:                     .filter(database.Order.custid == '222'))
40:                 row = query.one()
41:                 row.quantity += 1
42:
43:                 # Simulate a HW/SW failure occurring randomly.
44:                 if random.randrange(5) == 0:
45:                     print(f'Simulated failure with i = {i}')
46:                     sys.exit(1)
47:
48:                 query = (session.query(database.Book)
49:                     .filter(database.Book.isbn == '123'))
50:                 row = query.one()
51:                 row.quantity -= 1
52:
53:                 session.commit()
54:                 print('Transaction %d committed.' % i)
55:
56:     except Exception as ex:
57:         print(f'{sys.argv[0]}: {ex}', file=sys.stderr)
58:         sys.exit(1)
59:
60: #-----
61:
62: if __name__ == '__main__':
63:     main()

```