

pennyrender/database1.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2: #
3: #-----
4: # database.py
5: # Author: Bob Dondero
6: #-----
7:
8: import sqlite3
9: import contextlib
10:
11: #-----
12:
13: _DATABASE_URL = 'file:penny.sqlite?mode=ro'
14:
15: #-----
16:
17: def get_books(author):
18:
19:     books = []
20:
21:     with sqlite3.connect(_DATABASE_URL, isolation_level=None,
22:         uri=True) as connection:
23:
24:         with contextlib.closing(connection.cursor()) as cursor:
25:
26:             query_str = "SELECT isbn, author, title FROM books "
27:             query_str += "WHERE author LIKE ?"
28:             cursor.execute(query_str, [author+'%'])
29:
30:             table = cursor.fetchall()
31:             for row in table:
32:                 book = {'isbn': row[0], 'author': row[1],
33:                     'title': row[2]}
34:                 books.append(book)
35:
36:     return books
37:
38: #-----
39:
40: def _test():
41:
42:     books = get_books('ker')
43:     for book in books:
44:         print(book['isbn'])
45:         print(book['author'])
46:         print(book['title'])
47:         print()
48:
49: if __name__ == '__main__':
50:     _test()

```

pennyrender/database2.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2: #
3: #-----
4: # database.py
5: # Author: Bob Dondero
6: #-----
7:
8: import os
9: import psycpg2
10: import dotenv
11:
12: #-----
13:
14: dotenv.load_dotenv()
15: _DATABASE_URL = os.environ['DATABASE_URL']
16:
17: #-----
18:
19: def get_books(author):
20:
21:     books = []
22:
23:     with psycpg2.connect(_DATABASE_URL) as connection:
24:
25:         with connection.cursor() as cursor:
26:
27:             query_str = "SELECT isbn, author, title FROM books "
28:             query_str += "WHERE author ILIKE %s"
29:             cursor.execute(query_str, [author+'%'])
30:
31:             table = cursor.fetchall()
32:             for row in table:
33:                 book = {'isbn': row[0], 'author': row[1],
34:                     'title': row[2]}
35:                 books.append(book)
36:
37:     return books
38:
39: #-----
40:
41: # For testing:
42:
43: def _test():
44:     books = get_books('ker')
45:     for book in books:
46:         print(book['isbn'])
47:         print(book['author'])
48:         print(book['title'])
49:         print()
50:
51: if __name__ == '__main__':
52:     _test()

```

pennyrender/database3.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # database.py
5: # Author: Bob Dondero
6: #-----
7:
8: import os
9: import queue
10: import psycopg2
11: import dotenv
12:
13: #-----
14:
15: dotenv.load_dotenv()
16: _DATABASE_URL = os.environ['DATABASE_URL']
17:
18: _connection_pool = queue.Queue()
19:
20: #-----
21:
22: def _get_connection():
23:     try:
24:         conn = _connection_pool.get(block=False)
25:     except queue.Empty:
26:         conn = psycopg2.connect(_DATABASE_URL)
27:     return conn
28:
29: def _put_connection(conn):
30:     _connection_pool.put(conn)
31:
32: #-----
33:
34: def get_books(author):
35:     books = []
36:     connection = _get_connection()
37:     try:
38:         with connection.cursor() as cursor:
39:             query_str = "SELECT isbn, author, title FROM books "
40:             query_str += "WHERE author ILIKE %s"
41:             cursor.execute(query_str, [author+'%'])
42:             table = cursor.fetchall()
43:             for row in table:
44:                 book = {'isbn': row[0], 'author': row[1],
45:                        'title': row[2]}
46:                 books.append(book)
47:     finally:
48:         _put_connection(connection)
49:
50:     return books
51:
52: #-----
53:
54: # For testing:
55:
56: def _test():
57:     books = get_books('ker')
58:     for book in books:
59:         print(book['isbn'])
60:         print(book['author'])
61:         print(book['title'])
62:     print()
63:
64: if __name__ == '__main__':
65:     _test()

```

pennyrender/database.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # database.py
5: # Author: Bob Dondero
6: #-----
7:
8: import os
9: import sqlalchemy
10: import sqlalchemy.orm
11: import dotenv
12:
13: #-----
14:
15: dotenv.load_dotenv()
16: _DATABASE_URL = os.environ['DATABASE_URL']
17: _DATABASE_URL = _DATABASE_URL.replace('postgres://', 'postgresql://')
18:
19: #-----
20:
21: class Base(sqlalchemy.orm.DeclarativeBase):
22:     pass
23:
24: class Book(Base):
25:     __tablename__ = 'books'
26:     isbn = sqlalchemy.Column(sqlalchemy.String, primary_key=True)
27:     author = sqlalchemy.Column(sqlalchemy.String)
28:     title = sqlalchemy.Column(sqlalchemy.String)
29:
30:     _engine = sqlalchemy.create_engine(_DATABASE_URL)
31:
32: #-----
33:
34: def get_books(author):
35:
36:     books = []
37:
38:     with sqlalchemy.orm.Session(_engine) as session:
39:
40:         query = session.query(Book).filter(
41:             Book.author.ilike(author+'%'))
42:         table = query.all()
43:         for row in table:
44:             book = {'isbn': row.isbn, 'author': row.author,
45:                    'title': row.title}
46:             books.append(book)
47:
48:     return books
49:
50: #-----
51:
52: # For testing:
53:
54: def _test():
55:     books = get_books('ker')
56:     for book in books:
57:         print(book['isbn'])
58:         print(book['author'])
59:         print(book['title'])
60:     print()
61:
62: if __name__ == '__main__':
63:     _test()

```