

PostgreSQL/create.py (Page 1 of 3)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # create.py
5: # Author: Bob Dondero
6: #-----
7:
8: import os
9: import sys
10: import psycopg
11: import dotenv
12:
13: dotenv.load_dotenv()
14: DATABASE_URL = os.environ['DATABASE_URL']
15:
16: #-----
17:
18: def main():
19:
20:     if len(sys.argv) != 1:
21:         print(f'usage: python {sys.argv[0]}', file=sys.stderr)
22:         sys.exit(1)
23:
24:     try:
25:         with psycopg.connect(DATABASE_URL) as connection:
26:             with connection.cursor() as cursor:
27:
28:                 cursor.execute('DROP TABLE IF EXISTS orders')
29:                 cursor.execute('DROP TABLE IF EXISTS customers')
30:                 cursor.execute('DROP TABLE IF EXISTS zipcodes')
31:                 cursor.execute('DROP TABLE IF EXISTS authors')
32:                 cursor.execute('DROP TABLE IF EXISTS books')
33:
34:             #-----
35:
36:             cursor.execute('''
37:                 CREATE TABLE books (
38:                     isbn TEXT NOT NULL,
39:                     title TEXT,
40:                     quantity INTEGER,
41:                     PRIMARY KEY (isbn))
42:                 ''')
43:
44:             cursor.execute('''
45:                 CREATE TABLE authors (
46:                     isbn TEXT NOT NULL,
47:                     author TEXT NOT NULL,
48:                     PRIMARY KEY (isbn, author),
49:                     FOREIGN KEY (isbn) REFERENCES books(isbn))
50:                 ''')
51:
52:             cursor.execute('''
53:                 CREATE TABLE zipcodes (
54:                     zipcode TEXT NOT NULL,
55:                     city TEXT,
56:                     state TEXT,
57:                     PRIMARY KEY (zipcode))
58:                 ''')
59:
60:             cursor.execute('''
61:                 CREATE TABLE customers (
62:                     custid TEXT NOT NULL,
63:                     custname TEXT,
64:                     street TEXT,
65:                     zipcode TEXT,
66:                     PRIMARY KEY (custid),
67:                     FOREIGN KEY (zipcode)
68:                         REFERENCES zipcodes(zipcode))'''

```

PostgreSQL/create.py (Page 2 of 3)

```

66:         cursor.execute('''
67:             CREATE TABLE orders (
68:                 isbn TEXT NOT NULL,
69:                 custid TEXT NOT NULL,
70:                 quantity INTEGER,
71:                 PRIMARY KEY (isbn, custid),
72:                 FOREIGN KEY (isbn) REFERENCES books(isbn),
73:                 FOREIGN KEY (custid)
74:                     REFERENCES customers(custid))
75:             ''')
76:
77:         #-----
78:
79:         cursor.execute('''
80:             INSERT INTO books (isbn, title, quantity)
81:             VALUES ('123','The Practice of Programming',500)
82:             ''')
83:
84:         cursor.execute('''
85:             INSERT INTO books (isbn, title, quantity)
86:             VALUES ('234','The C Programming Language',800)
87:             ''')
88:
89:         cursor.execute('''
90:             INSERT INTO books (isbn, title, quantity)
91:             VALUES ('345','Algorithms in C',650)
92:             ''')
93:
94:         #-----
95:
96:         cursor.execute('''INSERT INTO authors (isbn, author)
97:             VALUES ('123','Kernighan')''')
98:
99:         cursor.execute('''INSERT INTO authors (isbn, author)
100:             VALUES ('123','Pike')''')
101:
102:         cursor.execute('''INSERT INTO authors (isbn, author)
103:             VALUES ('234','Kernighan')''')
104:
105:         cursor.execute('''INSERT INTO authors (isbn, author)
106:             VALUES ('234','Ritchie')''')
107:
108:         cursor.execute('''INSERT INTO authors (isbn, author)
109:             VALUES ('345','Sedgewick')''')
110:
111:         #-----
112:
113:         cursor.execute('''
114:             INSERT INTO zipcodes (zipcode, city, state)
115:             VALUES ('08540','Princeton','NJ')''')
116:
117:         cursor.execute('''
118:             INSERT INTO zipcodes (zipcode, city, state)
119:             VALUES ('02138','Cambridge','MA')''')
120:
121:         cursor.execute('''
122:             INSERT INTO zipcodes (zipcode, city, state)
123:             VALUES ('02142','Cambridge','MA')''')
124:
125:         #-----
126:
127:         cursor.execute('''
128:             INSERT INTO customers
129:             (custid, custname, street, zipcode) VALUES
130:             ('111','Princeton','114 Nassau St','08540')''')
131:
132:         cursor.execute('''
133:             INSERT INTO customers
134:             (custid, custname, street, zipcode) VALUES
135:             ('222','Harvard','1256 Mass Ave','02138')''')
136:
137:         cursor.execute('''
138:             INSERT INTO customers
139:             (custid, custname, street, zipcode) VALUES
140:             ('333','MIT','292 Main St','02142')''')

```

PostgreSQL/create.py (Page 3 of 3)

```

131:
132:      #-----
133:
134:      cursor.execute('''
135:          INSERT INTO orders (isbn, custid, quantity)
136:          VALUES ('123','222',20)''' )
137:      cursor.execute('''
138:          INSERT INTO orders (isbn, custid, quantity)
139:          VALUES ('345','222',100)''' )
140:      cursor.execute('''
141:          INSERT INTO orders (isbn, custid, quantity)
142:          VALUES ('123','111',30)''' )
143:
144:      #-----
145:
146:      connection.commit()
147:
148:      #-----
149:
150:  except Exception as ex:
151:      print(f'{sys.argv[0]}: {ex}', file=sys.stderr)
152:      sys.exit(1)
153:
154:  #-----
155:
156:  if __name__ == '__main__':
157:      main()

```

blank (Page 1 of 1)

1: This page is intentionally blank.

PostgreSQL/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 psycopg
11: import dotenv
12:
13: dotenv.load_dotenv()
14: DATABASE_URL = os.environ['DATABASE_URL']
15:
16: #-----
17:
18: def main():
19:
20:     if len(sys.argv) != 1:
21:         print(f'usage: python {sys.argv[0]}', file=sys.stderr)
22:         sys.exit(1)
23:
24:     try:
25:         with psycopg.connect(DATABASE_URL) as connection:
26:             with connection.cursor() as cursor:
27:
28:                 print('-----')
29:                 print('books')
30:                 print('-----')
31:                 cursor.execute("SELECT * FROM books")
32:                 table = cursor.fetchall()
33:                 for row in table:
34:                     print(row)
35:
36:                 print('-----')
37:                 print('authors')
38:                 print('-----')
39:                 cursor.execute("SELECT * FROM authors")
40:                 table = cursor.fetchall()
41:                 for row in table:
42:                     print(row)
43:
44:                 print('-----')
45:                 print('zipcodes')
46:                 print('-----')
47:                 cursor.execute("SELECT * FROM zipcodes")
48:                 table = cursor.fetchall()
49:                 for row in table:
50:                     print(row)
51:
52:                 print('-----')
53:                 print('customers')
54:                 print('-----')
55:                 cursor.execute("SELECT * FROM customers")
56:                 table = cursor.fetchall()
57:                 for row in table:
58:                     print(row)
59:
60:                 print('-----')
61:                 print('orders')
62:                 print('-----')
63:                 cursor.execute("SELECT * FROM orders")
64:                 table = cursor.fetchall()
65:                 for row in table:

```

PostgreSQL/display.py (Page 2 of 2)

```

66:                 print(row)
67:
68:     except Exception as ex:
69:         print(f'{sys.argv[0]}: {ex}', file=sys.stderr)
70:         sys.exit(1)
71:
72: #-----
73:
74: if __name__ == '__main__':
75:     main()

```

PostgreSQL/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 psycopg
11: import dotenv
12:
13: dotenv.load_dotenv()
14: DATABASE_URL = os.environ['DATABASE_URL']
15:
16: #-----
17:
18: def main():
19:
20:     if len(sys.argv) != 2:
21:         print(f'usage: python {sys.argv[0]} author', file=sys.stderr)
22:         sys.exit(1)
23:
24:     author = sys.argv[1]
25:
26:     try:
27:         with psycopg.connect(DATABASE_URL) as connection:
28:             with connection.cursor() as cursor:
29:                 stmt_str = f'''
30:                     SELECT books.isbn, title, quantity
31:                     FROM books, authors
32:                     WHERE books.isbn = authors.isbn
33:                     AND author = '{author}'
34:                     '''
35:                 cursor.execute(stmt_str)
36:                 table = cursor.fetchall()
37:                 for row in table:
38:                     print('ISBN:', row[0])
39:                     print('Title:', row[1])
40:                     print('Quantity:', row[2])
41:                     print()
42:
43:     except Exception as ex:
44:         print(f'{sys.argv[0]}: {ex}', file=sys.stderr)
45:         sys.exit(1)
46:
47: #-----
48:
49: if __name__ == '__main__':
50:     main()

```

PostgreSQL/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 psycopg
11: import dotenv
12:
13: dotenv.load_dotenv()
14: DATABASE_URL = os.environ['DATABASE_URL']
15:
16: #-----
17:
18:
19: def main():
20:
21:     if len(sys.argv) != 3:
22:         print(f'usage: python {sys.argv[0]} isbn custid',
23:             file=sys.stderr)
24:         sys.exit(1)
25:
26:     isbn = sys.argv[1]
27:     custid = sys.argv[2]
28:
29:     try:
30:         with psycopg.connect(DATABASE_URL) as connection:
31:             with connection.cursor() as cursor:
32:                 stmt_str = f'''
33:                     UPDATE orders SET quantity = quantity+1
34:                     WHERE isbn = '{isbn}' AND custid = '{custid}'
35:                     '''
36:                 cursor.execute(stmt_str)
37:                 connection.commit()
38:
39:     except Exception as ex:
40:         print(f'{sys.argv[0]}: {ex}', file=sys.stderr)
41:         sys.exit(1)
42:
43: #-----
44:
45: if __name__ == '__main__':
46:     main()

```

PostgreSQL/authorsearchprep.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # authorsearchprep.py
5: # Author: Bob Dondero
6: #-----
7:
8: import os
9: import sys
10: import psycopg
11: import dotenv
12:
13: dotenv.load_dotenv()
14: DATABASE_URL = os.environ['DATABASE_URL']
15:
16: #-----
17:
18: def main():
19:
20:     if len(sys.argv) != 2:
21:         print(f'usage: python {sys.argv[0]} author', file=sys.stderr)
22:         sys.exit(1)
23:
24:     author = sys.argv[1]
25:
26:     try:
27:         with psycopg.connect(DATABASE_URL) as connection:
28:             with connection.cursor() as cursor:
29:                 stmt_str = '''
30:                     SELECT books.isbn, title, quantity
31:                     FROM books, authors
32:                     WHERE books.isbn = authors.isbn
33:                     AND author = %s
34:                     '''
35:                 cursor.execute(stmt_str, [author])
36:                 table = cursor.fetchall()
37:                 for row in table:
38:                     print('ISBN:', row[0])
39:                     print('Title:', row[1])
40:                     print('Quantity:', row[2])
41:                     print()
42:
43:     except Exception as ex:
44:         print(f'{sys.argv[0]}: {ex}', file=sys.stderr)
45:         sys.exit(1)
46:
47: #-----
48:
49: if __name__ == '__main__':
50:     main()

```

PostgreSQL/orderprep.py (Page 1 of 1)

```

1: #!/usr/bin/env python
2:
3: #-----
4: # orderprep.py
5: # Author: Bob Dondero
6: #-----
7:
8: import os
9: import sys
10: import psycopg
11: import dotenv
12:
13: dotenv.load_dotenv()
14: DATABASE_URL = os.environ['DATABASE_URL']
15:
16: #-----
17:
18:
19: def main():
20:
21:     if len(sys.argv) != 3:
22:         print(f'usage: python {sys.argv[0]} isbn custid',
23:             file=sys.stderr)
24:         sys.exit(1)
25:
26:     isbn = sys.argv[1]
27:     custid = sys.argv[2]
28:
29:     try:
30:         with psycopg.connect(DATABASE_URL) as connection:
31:             with connection.cursor() as cursor:
32:                 stmt_str = '''
33:                     UPDATE orders SET quantity = quantity+1
34:                     WHERE isbn = %s AND custid = %s
35:                     '''
36:                 cursor.execute(stmt_str, [isbn, custid])
37:                 connection.commit()
38:
39:     except Exception as ex:
40:         print(f'{sys.argv[0]}: {ex}', file=sys.stderr)
41:         sys.exit(1)
42:
43: #-----
44:
45: if __name__ == '__main__':
46:     main()

```

PostgreSQL/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 psycopg
11: import dotenv
12:
13: dotenv.load_dotenv()
14: DATABASE_URL = os.environ['DATABASE_URL']
15:
16: #-----
17:
18: def main():
19:
20:     if len(sys.argv) != 3:
21:         print(f'usage: python {sys.argv[0]} isbn custid',
22:               file=sys.stderr)
23:         sys.exit(1)
24:
25:     isbn = sys.argv[1]
26:     custid = sys.argv[2]
27:
28:     try:
29:         with psycopg.connect(DATABASE_URL) as connection:
30:             with connection.cursor() as cursor:
31:                 stmt_str = '''
32:                     UPDATE orders SET quantity = quantity+1
33:                     WHERE isbn = %s AND custid = %s
34:                     '''
35:                 cursor.execute(stmt_str, [isbn, custid])
36:                 stmt_str = '''
37:                     UPDATE books SET quantity = quantity-1
38:                     WHERE isbn = %s
39:                     '''
40:                 cursor.execute(stmt_str, [isbn])
41:                 connection.commit()
42:                 print('Transaction committed.')
43:
44:     except Exception as ex:
45:         print(f'{sys.argv[0]}: {ex}', file=sys.stderr)
46:         sys.exit(1)
47:
48: #-----
49:
50: if __name__ == '__main__':
51:     main()

```

PostgreSQL/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 psycopg
12: import dotenv
13:
14: dotenv.load_dotenv()
15: DATABASE_URL = os.environ['DATABASE_URL']
16:
17: #-----
18:
19: def main():
20:
21:     if len(sys.argv) != 1:
22:         print(f'usage: python {sys.argv[0]}', file=sys.stderr)
23:         sys.exit(1)
24:
25:     try:
26:         connection = psycopg.connect(DATABASE_URL)
27:         cursor = connection.cursor()
28:
29:         for i in range(20):
30:             stmt_str = '''
31:                 UPDATE orders SET quantity = quantity+1
32:                 WHERE isbn = '123' AND custid = '222'
33:                 '''
34:             cursor.execute(stmt_str)
35:
36:             # Simulate a HW/SW failure occurring randomly.
37:             if random.randrange(10) == 0:
38:                 print(f'Simulated failure with i = {i}')
39:                 sys.exit(1)
40:
41:             stmt_str = '''
42:                 UPDATE books SET quantity = quantity-1
43:                 WHERE isbn = '123'
44:                 '''
45:             cursor.execute(stmt_str)
46:             connection.commit()
47:             print('Transaction %d committed.' % i)
48:
49:         cursor.close()
50:         connection.close()
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()

```