

PostgreSQL/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 pycpg2
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('Usage: python ' + sys.argv[0], file=sys.stderr)
22:         sys.exit(1)
23:
24:     try:
25:         with pycpg2.connect(DATABASE_URL) as connection:
26:
27:             with connection.cursor() as cursor:
28:
29:                 #-----
30:
31:                 cursor.execute("DROP TABLE IF EXISTS books")
32:                 cursor.execute("CREATE TABLE books "
33:                     + "(isbn TEXT, title TEXT, quantity INTEGER)")
34:                 cursor.execute("INSERT INTO books "
35:                     + "(isbn, title, quantity) "
36:                     + "VALUES ('123',"
37:                     + "'The Practice of Programming',500)")
38:                 cursor.execute("INSERT INTO books "
39:                     + "(isbn, title, quantity) "
40:                     + "VALUES ('234',"
41:                     + "'The C Programming Language',800)")
42:                 cursor.execute("INSERT INTO books "
43:                     + "(isbn, title, quantity) "
44:                     + "VALUES ('345',"
45:                     + "'Algorithms in C',650)")
46:
47:                 #-----
48:
49:                 cursor.execute("DROP TABLE IF EXISTS authors")
50:                 cursor.execute("CREATE TABLE authors "
51:                     + "(isbn TEXT, author TEXT)")
52:                 cursor.execute("INSERT INTO authors (isbn, author) "
53:                     + "VALUES ('123','Kernighan')")
54:                 cursor.execute("INSERT INTO authors (isbn, author) "
55:                     + "VALUES ('123','Pike')")
56:                 cursor.execute("INSERT INTO authors (isbn, author) "
57:                     + "VALUES ('234','Kernighan')")
58:                 cursor.execute("INSERT INTO authors (isbn, author) "
59:                     + "VALUES ('234','Ritchie')")
60:                 cursor.execute("INSERT INTO authors (isbn, author) "
61:                     + "VALUES ('345','Sedgewick')")
62:
63:                 #-----
64:
65:                 cursor.execute("DROP TABLE IF EXISTS customers")

```

PostgreSQL/create.py (Page 2 of 2)

```

66:         cursor.execute("CREATE TABLE customers "
67:             + "(custid TEXT, custname TEXT, street TEXT, "
68:             + "zipcode TEXT)")
69:         cursor.execute("INSERT INTO customers "
70:             + "(custid, custname, street, zipcode) VALUES "
71:             + "('111','Princeton','114 Nassau St','08540')")
72:         cursor.execute("INSERT INTO customers "
73:             + "(custid, custname, street, zipcode) VALUES "
74:             + "('222','Harvard','1256 Mass Ave','02138')")
75:         cursor.execute("INSERT INTO customers "
76:             + "(custid, custname, street, zipcode) VALUES "
77:             + "('333','MIT','292 Main St','02142')")
78:
79:         #-----
80:
81:         cursor.execute("DROP TABLE IF EXISTS zipcodes")
82:         cursor.execute("CREATE TABLE zipcodes "
83:             + "(zipcode TEXT, city TEXT, state TEXT)")
84:         cursor.execute("INSERT INTO zipcodes "
85:             + "(zipcode, city, state) "
86:             + "VALUES ('08540','Princeton','NJ')")
87:         cursor.execute("INSERT INTO zipcodes "
88:             + "(zipcode, city, state) "
89:             + "VALUES ('02138','Cambridge','MA')")
90:         cursor.execute("INSERT INTO zipcodes "
91:             + "(zipcode, city, state) "
92:             + "VALUES ('02142','Cambridge','MA')")
93:
94:         #-----
95:
96:         cursor.execute("DROP TABLE IF EXISTS orders")
97:         cursor.execute("CREATE TABLE orders "
98:             + "(isbn TEXT, custid TEXT, quantity INTEGER)")
99:         cursor.execute("INSERT INTO orders (isbn, custid, "
100:             + "quantity) "
101:             + "VALUES ('123','222',20)")
102:         cursor.execute("INSERT INTO orders (isbn, custid, "
103:             + "quantity) "
104:             + "VALUES ('345','222',100)")
105:         cursor.execute("INSERT INTO orders (isbn, custid, "
106:             + "quantity) "
107:             + "VALUES ('123','111',30)")
108:
109:         #-----
110:
111:     except Exception as ex:
112:         print(ex, file=sys.stderr)
113:         sys.exit(1)
114:
115:     #-----
116:
117: if __name__ == '__main__':
118:     main()

```

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 psycopg2
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('Usage: python ' + sys.argv[0], file=sys.stderr)
22:         sys.exit(1)
23:
24:     try:
25:         with psycopg2.connect(DATABASE_URL) as connection:
26:
27:             with connection.cursor() as cursor:
28:
29:                 print('-----')
30:                 print('books')
31:                 print('-----')
32:                 cursor.execute("SELECT * FROM books")
33:                 table = cursor.fetchall()
34:                 for row in table:
35:                     print(row)
36:
37:                 print('-----')
38:                 print('authors')
39:                 print('-----')
40:                 cursor.execute("SELECT * FROM authors")
41:                 table = cursor.fetchall()
42:                 for row in table:
43:                     print(row)
44:
45:                 print('-----')
46:                 print('customers')
47:                 print('-----')
48:                 cursor.execute("SELECT * FROM customers")
49:                 table = cursor.fetchall()
50:                 for row in table:
51:                     print(row)
52:
53:                 print('-----')
54:                 print('zipcodes')
55:                 print('-----')
56:                 cursor.execute("SELECT * FROM zipcodes")
57:                 table = cursor.fetchall()
58:                 for row in table:
59:                     print(row)
60:
61:                 print('-----')
62:                 print('orders')
63:                 print('-----')
64:                 cursor.execute("SELECT * FROM orders")
65:                 table = cursor.fetchall()

```

PostgreSQL/display.py (Page 2 of 2)

```

66:                 for row in table:
67:                     print(row)
68:
69:             except Exception as ex:
70:                 print(ex, file=sys.stderr)
71:                 sys.exit(1)
72:
73: #-----
74:
75: if __name__ == '__main__':
76:     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 pycpg2
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('Usage: python ' + sys.argv[0] + ' author',
22:               file=sys.stderr)
23:         sys.exit(1)
24:
25:     author = sys.argv[1]
26:
27:     try:
28:         with pycpg2.connect(DATABASE_URL) as connection:
29:
30:             with connection.cursor() as cursor:
31:
32:                 stmt_str = 'SELECT books.isbn, title, quantity '
33:                 stmt_str += 'FROM books, authors '
34:                 stmt_str += 'WHERE books.isbn = authors.isbn '
35:                 stmt_str += 'AND author = \'' + author + '\''
36:                 cursor.execute(stmt_str)
37:
38:                 table = cursor.fetchall()
39:                 for row in table:
40:                     print('ISBN:', row[0])
41:                     print('Title:', row[1])
42:                     print('Quantity:', row[2])
43:                     print()
44:
45:     except Exception as ex:
46:         print(ex, file=sys.stderr)
47:         sys.exit(1)
48:
49: #-----
50:
51: if __name__ == '__main__':
52:     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 pycpg2
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('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 pycpg2.connect(DATABASE_URL) as connection:
31:
32:             with connection.cursor() as cursor:
33:
34:                 stmt_str = "UPDATE orders SET quantity = quantity+1 "
35:                 stmt_str += "WHERE isbn = '" + isbn
36:                 stmt_str += "' AND custid = '" + custid + "'"
37:                 cursor.execute(stmt_str)
38:
39:     except Exception as ex:
40:         print(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 psycopg2
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('Usage: python ' + sys.argv[0] + ' author',
22:               file=sys.stderr)
23:         sys.exit(1)
24:
25:     author = sys.argv[1]
26:
27:     try:
28:         with psycopg2.connect(DATABASE_URL) as connection:
29:
30:             with connection.cursor() as cursor:
31:
32:                 # Create a prepared statement and substitute values.
33:                 stmt_str = 'SELECT books.isbn, title, quantity '
34:                 stmt_str += 'FROM books, authors '
35:                 stmt_str += 'WHERE books.isbn = authors.isbn '
36:                 stmt_str += 'AND author = %s'
37:                 cursor.execute(stmt_str, [author])
38:
39:                 table = cursor.fetchall()
40:                 for row in table:
41:                     print('ISBN:', row[0])
42:                     print('Title:', row[1])
43:                     print('Quantity:', row[2])
44:                     print()
45:
46:     except Exception as ex:
47:         print(ex, file=sys.stderr)
48:         sys.exit(1)
49:
50: #-----
51:
52: if __name__ == '__main__':
53:     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 psycopg2
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('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 psycopg2.connect(DATABASE_URL) as connection:
31:
32:             with connection.cursor() as cursor:
33:
34:                 stmt_str = "UPDATE orders SET quantity = quantity+1 "
35:                 stmt_str += "WHERE isbn = %s AND custid = %s"
36:                 cursor.execute(stmt_str, [isbn, custid])
37:
38:     except Exception as ex:
39:         print(ex, file=sys.stderr)
40:         sys.exit(1)
41:
42: #-----
43:
44: if __name__ == '__main__':
45:     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 psycpg2
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('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 psycpg2.connect(DATABASE_URL) as connection:
30:
31:             with connection.cursor() as cursor:
32:
33:                 cursor.execute('BEGIN')
34:
35:                 stmt_str = "UPDATE orders SET quantity = quantity+1 "
36:                 stmt_str += "WHERE isbn = %s AND custid = %s"
37:                 cursor.execute(stmt_str, [isbn, custid])
38:
39:                 stmt_str = "UPDATE books SET quantity = quantity-1 "
40:                 stmt_str += "WHERE isbn = %s"
41:                 cursor.execute(stmt_str, [isbn])
42:
43:                 cursor.execute('COMMIT')
44:                 print('Transaction committed.')
45:
46:     except Exception as ex:
47:         print(ex, file=sys.stderr)
48:         sys.exit(1)
49:
50: #-----
51:
52: if __name__ == '__main__':
53:     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 psycpg2
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('Usage: python ' + sys.argv[0], file=sys.stderr)
23:         sys.exit(1)
24:
25:     try:
26:         with psycpg2.connect(DATABASE_URL) as connection:
27:
28:             with connection.cursor() as cursor:
29:
30:                 for i in range(20):
31:
32:                     cursor.execute('BEGIN')
33:
34:                     stmt_str = "UPDATE orders "
35:                     stmt_str += "SET quantity = quantity+1 "
36:                     stmt_str += "WHERE isbn = '123' "
37:                     stmt_str += "AND custid = '222'"
38:                     cursor.execute(stmt_str)
39:
40:                     # Simulate a HW/SW failure occurring randomly,
41:                     # on average every 5th time through the loop.
42:                     if random.randrange(5) == 0:
43:                         print('Simulated failure with i = %d' % i)
44:                         cursor.execute('ROLLBACK')
45:                         print('Transaction rolled back.')
46:                         continue
47:
48:                     stmt_str = "UPDATE books "
49:                     stmt_str += "SET quantity = quantity-1 "
50:                     stmt_str += "WHERE isbn = '123'"
51:                     cursor.execute(stmt_str)
52:
53:                     cursor.execute('COMMIT')
54:                     print('Transaction %d committed.' % i)
55:
56:     except Exception as ex:
57:         print(ex, file=sys.stderr)
58:         sys.exit(1)
59:
60: #-----
61:
62: if __name__ == '__main__':
63:     main()

```