```
import calendar
import cPickle
import datetime
import socket
import sqlite3
import sys

from bfcslib.sensor_reading import SensorReading

DB_PATH = '/home/dsorber/bfcs/beer_temps.db'
HOST = '192.168.1.1'
PORT = 2243

def main():

    # Create socket and wait for connections
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.bind((HOST, PORT))
    s.listen(1)
    conn, addr = s.accept()
    print 'Connected by', addr

    # Main loop
    while True:

        # Block and wait to receive data
        data = conn.recv(65536)

        # Try to unpickle any data received
        try:
            obj = cPickle.loads(data)
        except:
            print 'I got some data that I couldn\'t unpickle...'
            print '-' * 100
            print data
            print '-' * 100
            print '\n\n'
            continue

        # Is the unpickled object a SensorReading object?
        if isinstance(obj, SensorReading):
            print obj.time.strftime('%m/%d/%Y %H:%M:%S')
            print calendar.timegm(obj.time.timetuple())
            print '\tTemp 1: %s' % obj.temp1
            print '\tTemp 2: %s' % obj.temp2
            print '\tTemp 3: %s' % obj.temp3
            print '\tTemp 4: %s' % obj.temp4
            print '\tRelay is %s' % (obj.relay_status and 'on' or 'off')

            # Insert the reading data into a database
            print 'Inserting data into the database...'
            sys.stdout.flush()

            db_conn = sqlite3.connect(DB_PATH)
            db = db_conn.cursor()
            try:
                db.execute("INSERT INTO temp_data VALUES (%s, %s, %s, %s, %s, %s)" %
                           (calendar.timegm(obj.time.timetuple()), obj.temp1,
                            obj.temp2, obj.temp3, obj.temp4,
                            int(obj.relay_status and '1' or '0')))
                db_conn.commit()
            except:
                print 'FAILED'
                db_conn.close()
                continue

            db_conn.close()
            print 'SUCCESS'

    conn.close()


if __name__ == '__main__':
    sys.exit(main())
```