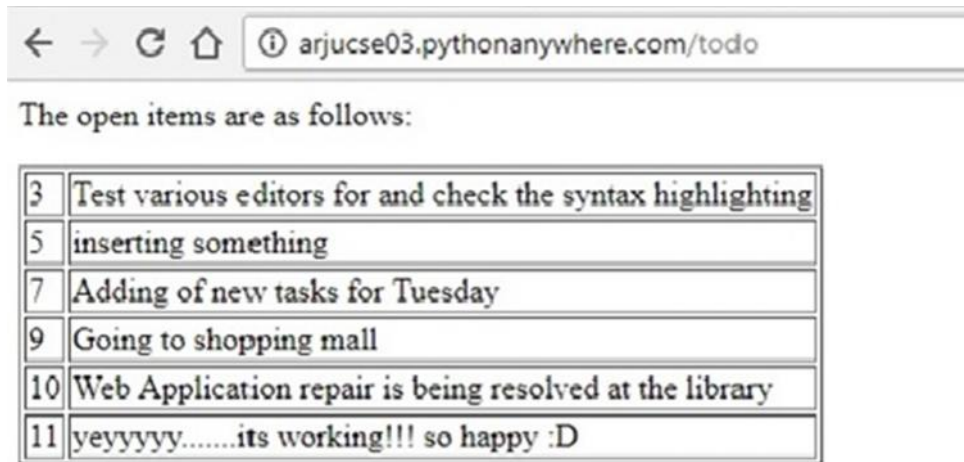


### 1. Web app running with SQL



### 2. Chinook database queries that join at least three tables, with and without custom indexes.

#### Without custom index:

Select distinct playlist.name from playlist, playlisttrack, track  
Where playlist.playlistid = playlisttrack.playlistid  
AND track.trackid = playlisttrack.trackid

#### With custom index:

Create unique index name ON Playlist (PlaylistId, Name);

Select distinct playlist.name from playlist, playlisttrack, track  
Where playlist.playlistid = playlisttrack.playlistid  
AND track.trackid = playlisttrack.trackid

### 3. Web app running with Mongo on MLAB

```
from pymongo import MongoClient

class TaskList():

    db = None
    def __init__(self):
        self.client
=MongoClient("mongodb://msultana:msultana@ds033196.mlab.com:33196/test_
database")
        self.db = self.client['test_database']

    def update_text(self, id, text):
        self.db.todo.update_one({"id":id}, {"$set":{"text":text}})
```

```
def update_status(self, id, status):
    self.db.todo.update_one({"id":id}, {"$set":{"status":status}})

def delete_everything(self):
    self.db.todo.delete_many({})

def delete_task(self, id):
    self.db.todo.delete_one({"id":id})
```

← → ↻ 🏠 ⓘ localhost:8080/tasks

### Todo List

<a href="#">Add new task...</a>			
<a href="#">adding item 1</a>	<a href="#">Complete</a>	<a href="#">Delete</a>	1
<a href="#">adding item2</a>	<a href="#">Complete</a>	<a href="#">Delete</a>	1
<a href="#">adding item3</a>	<a href="#">Complete</a>	<a href="#">Delete</a>	1
<a href="#">adding item4</a>	<a href="#">Complete</a>	<a href="#">Delete</a>	1
<a href="#">adding item5</a>	<a href="#">Complete</a>	<a href="#">Delete</a>	1

## 4. Mongo running on local server

```
Command Prompt - "C:\Program Files\MongoDB\Server\3.6\bin\mongod.exe"
2017-12-15T19:12:31.836-0800 I STORAGE [initandlisten] WiredTiger message [1513393951:835758][7500:14
0712987144960], txn-recover: Recovering log 8 through 8
2017-12-15T19:12:32.488-0800 I CONTROL [initandlisten]
2017-12-15T19:12:32.488-0800 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for
the database.
2017-12-15T19:12:32.490-0800 I CONTROL [initandlisten] **          Read and write access to data and
configuration is unrestricted.
2017-12-15T19:12:32.491-0800 I CONTROL [initandlisten]
2017-12-15T19:12:32.492-0800 I CONTROL [initandlisten] ** WARNING: This server is bound to localhost.
2017-12-15T19:12:32.494-0800 I CONTROL [initandlisten] **          Remote systems will be unable to c
onnect to this server.
2017-12-15T19:12:32.495-0800 I CONTROL [initandlisten] **          Start the server with --bind_ip <a
ddress> to specify which IP
2017-12-15T19:12:32.497-0800 I CONTROL [initandlisten] **          addresses it should serve response
s from, or with --bind_ip_all to
2017-12-15T19:12:32.498-0800 I CONTROL [initandlisten] **          bind to all interfaces. If this be
havior is desired, start the
2017-12-15T19:12:32.501-0800 I CONTROL [initandlisten] **          server with --bind_ip 127.0.0.1 to
disable this warning.
2017-12-15T19:12:32.502-0800 I CONTROL [initandlisten]
2017-12-15T19:12:32.503-0800 I CONTROL [initandlisten]
2017-12-15T19:12:32.505-0800 I CONTROL [initandlisten] ** WARNING: The file system cache of this mach
ine is configured to be greater than 40%
of the total memory. This can lead to increased memory pressu
re and poor performance.
2017-12-15T19:12:32.506-0800 I CONTROL [initandlisten] See http://dochub.mongodb.org/core/wt-windows-
system-file-cache
2017-12-15T19:12:32.507-0800 I CONTROL [initandlisten]
2017-12-15T22:12:34.381-0500 I FTDC [initandlisten] Initializing full-time diagnostic data capture
with directory 'C:/data/db/diagnostic.data'
2017-12-15T22:12:34.411-0500 I NETWORK [initandlisten] waiting for connections on port 27017
```

5. A Mongo query aggregation example using the Restaurant database

```
db.restaurent.findOne()
{
  "_id" : ObjectId("5a2ae97f4f25d1ae488a9ede"),
  "address" : {
    "building" : "1007",
    "coord" : [
      -73.856077,
      40.848447
    ],
    "street" : "Morris Park Ave",
    "zipcode" : "10462"
  },
  "borough" : "Bronx",
  "cuisine" : "Bakery",
  "grades" : [
    {
      "date" : ISODate("2014-03-03T00:00:00Z"),
      "grade" : "A",
      "score" : 2
    },
    {
      "date" : ISODate("2013-09-11T00:00:00Z"),
      "grade" : "A",
      "score" : 6
    },
    {
      "date" : ISODate("2013-01-24T00:00:00Z"),
      "grade" : "A",
      "score" : 10
    },
    {
      "date" : ISODate("2011-11-23T00:00:00Z"),
      "grade" : "A",
      "score" : 9
    },
    {
      "date" : ISODate("2011-03-10T00:00:00Z"),
      "grade" : "B",
      "score" : 14
    }
  ],
}
```

```
    "name" : "Morris Park Bake Shop",  
    "restaurant_id" : "30075445"  
  }
```

```
db.restaurent.aggregate( [ { $match: {"cuisine" : "Bakery" }}] )
```

6. A Mongo map-reduce example using Restaurant database

```
db.restaurent.mapReduce(  
  function(){ emit( this.date, this.score ); },  
  function(key, values) { return Array.sum(values) },  
  {  
    Query: { grade: "A"},  
    Out: "score_total"  
  })
```