

IT7359

ADVANCE DATABASE SYSTEM

Covert ERD to Other Databases

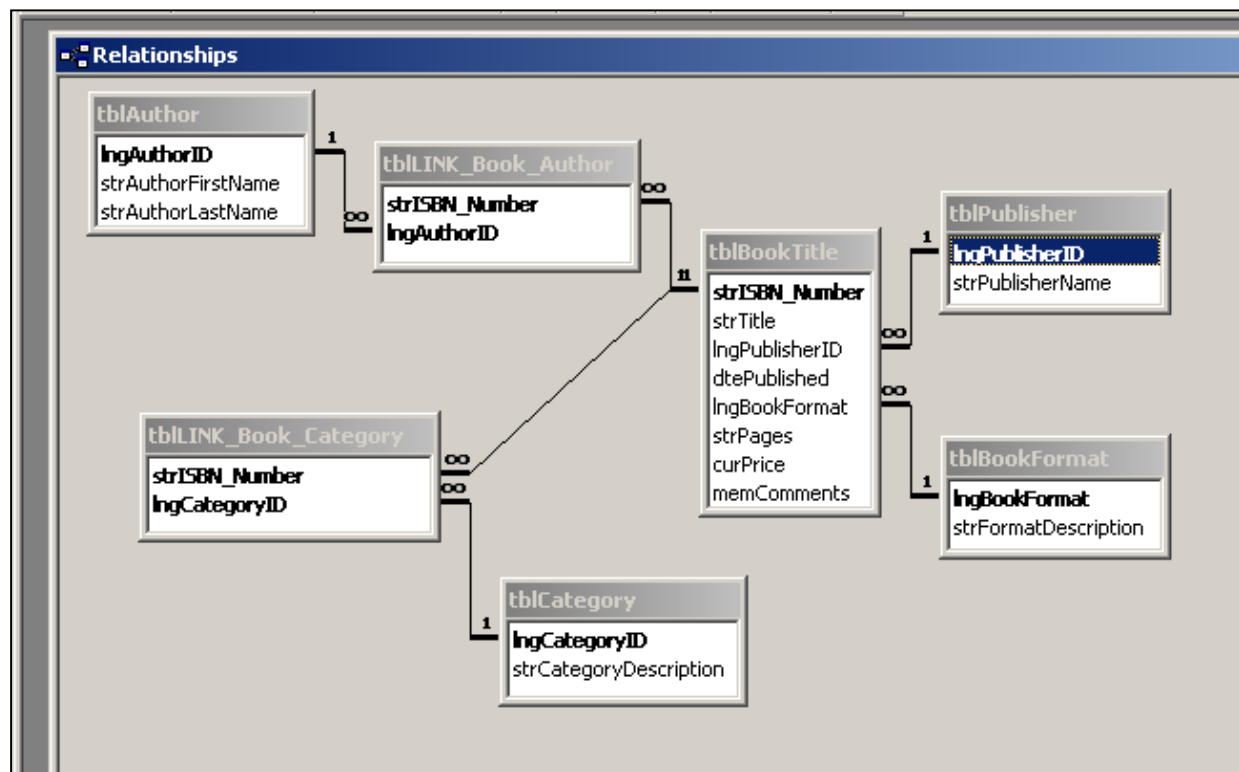
Table of Contents

ERD	4
Choosing database	5
Insert data	8
Query 1: List all the books Titles and their associated price	12
Query 2: List all the information of the books which have a price higher than 100\$ and list their corresponding author name and surnames	13
Query 3: Add a new book	14
Query 4: List all the books which belong to a specific category	15
Query 5: List all the author information ordered by their last name	16
Query 6: List the Title, Book Format (Description), Category (Description) and publisher name of all the books that contain the letter "P"	17
Query 7: Delete the book with the highest price	18

Table of Figures

Figure 1: MongoDB files.....	5
Figure 2: Mongo Compass	6
Figure 3: Mongod.....	6
Figure 4: Studio 3T	7
Figure 5: Database	11
Figure 6: Data type.....	11
Figure 7: Query 1.....	12
Figure 8: Query 2.....	13
Figure 9: Query 3.....	14
Figure 10: Query 4.....	15
Figure 11: Query 5.....	16
Figure 12: Query 6.....	17
Figure 13: Query 7 - highest value	18
Figure 14: deleteOne() function.....	18
Figure 15: book collection after deleting highest price book	19

ERD



Choosing database

As the given option, there are two databases that can choose to build the bookshop system in this assignment. Two of them have unique pros and cons. It can be said that, MongoDB stand out by its fast reading speed, can be built in short period. This project requires merely the PC version of the database so MongoDB will help the user to reduce the time of implement. Simple and fast to learn is another advance that MongoDB has for the syntax, especially for user get used to SQL. Base on their features, the picture, time and scope of this project, therefore MongoDB is chosen to be the database of this business.

Installing steps

After choosing the appropriate database, install file is downloaded and run. In this stage, to make it easier for the later process, its directory had been changed to a new place (which is created a folder in C drive):

C:\mongodb

After finish the installing, it can be seen that MongoDB has many files (figure 1), as the same time of installing process, Mongo Compass (figure 2), which is the coding application giving the interacting interface for MongoDB.

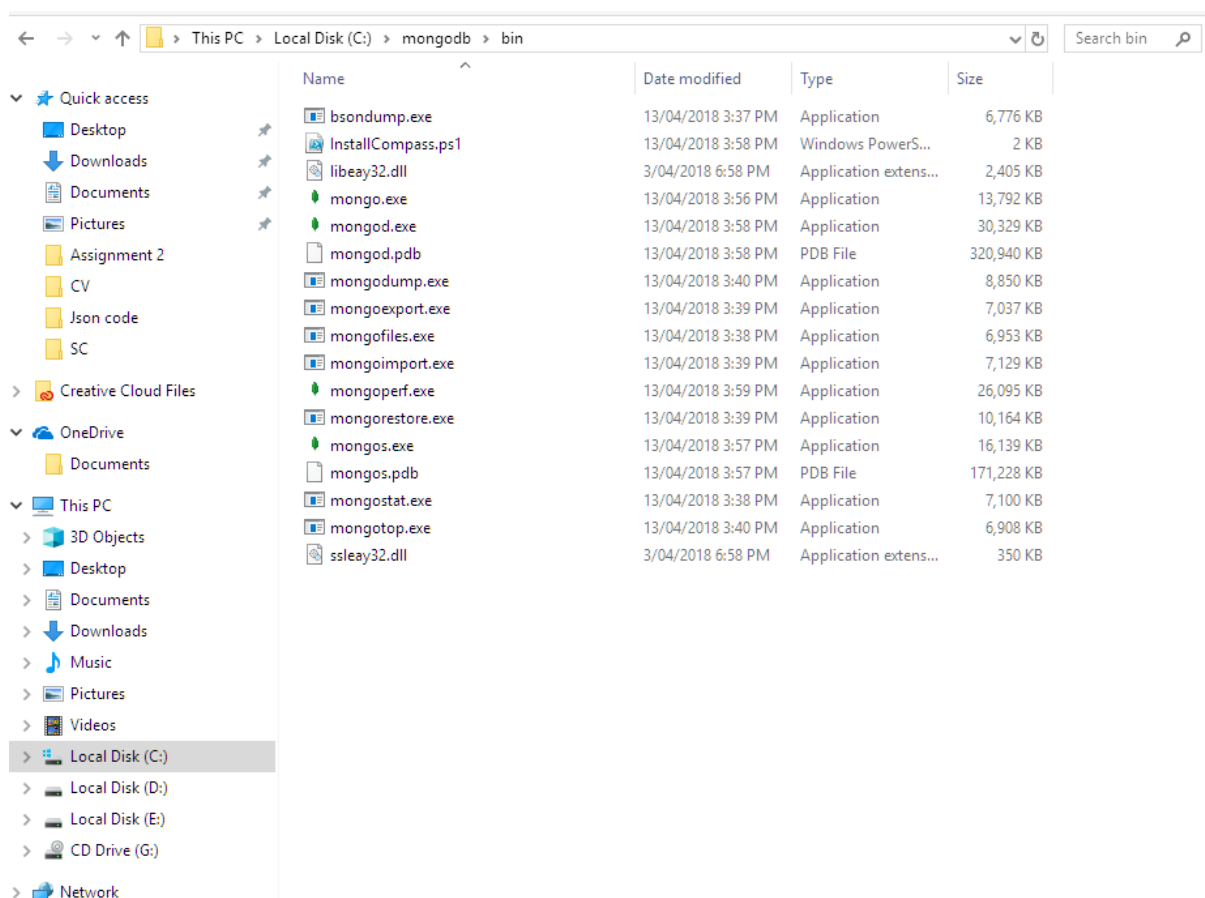


Figure 1: MongoDB files

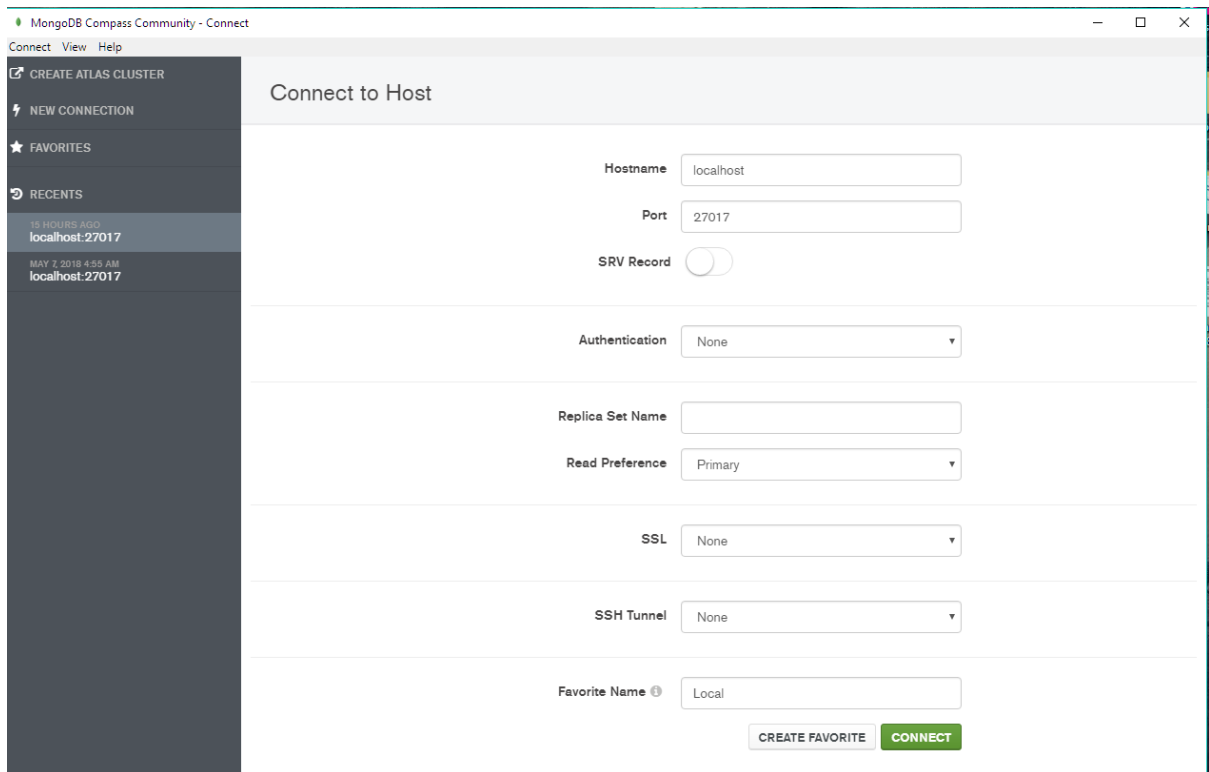


Figure 2: Mongo Compass

To make everything more comfortable, I already create the folder and change the path to let them store the database at C:\mogodb\data

To open the server, the application mongod.exe (figure 3) must be run. Then the database is connected, and we can start to build the database. Although MongoDB has provided Mongo Compass in the installing files, however, I choose Studio 3T (figure 4) as my working environment for MongoDB.

```

C:\Users\Will Pham\Desktop\mongodb-win32-x86_64-enterprise-windows-64-3.6.4\mongodb-win32-x86_64-enterprise-windows-64-3.6.4\bin\mongod...
./data/db/diagnostic.data'
2018-05-07T20:03:33.938+1200 I NETWORK [initandlisten] waiting for connections on port 27017
2018-05-07T20:03:39.419+1200 I NETWORK [listener] connection accepted from 127.0.0.1:55828 #1 (1 connection now open)
2018-05-07T20:03:39.420+1200 I NETWORK [conn1] received client metadata from 127.0.0.1:55828 conn1: { driver: { name: "
mongo-java-driver", version: "018.1.0-FINAL-104-gbf88a26" }, os: { type: "Windows", name: "Windows 10", architecture: "a
md64", version: "10.0" }, platform: "Java/Oracle Corporation/1.8.0_161-b12" }
2018-05-07T20:03:39.423+1200 I NETWORK [listener] connection accepted from 127.0.0.1:55829 #2 (2 connections now open)
2018-05-07T20:03:39.425+1200 I NETWORK [conn2] received client metadata from 127.0.0.1:55829 conn2: { driver: { name: "
mongo-java-driver", version: "018.1.0-FINAL-104-gbf88a26" }, os: { type: "Windows", name: "Windows 10", architecture: "a
md64", version: "10.0" }, platform: "Java/Oracle Corporation/1.8.0_161-b12" }
2018-05-07T20:03:46.513+1200 I NETWORK [listener] connection accepted from 127.0.0.1:55830 #3 (3 connections now open)
2018-05-07T20:03:46.516+1200 I NETWORK [conn3] received client metadata from 127.0.0.1:55830 conn3: { application: { na
me: "MongoDB Shell" }, driver: { name: "MongoDB Internal Client", version: "3.6.2" }, os: { type: "Windows", name: "Micr
osoft Windows 10", architecture: "x86_64", version: "10.0 (build 16299)" } }
2018-05-07T21:13:04.955+1200 I NETWORK [listener] connection accepted from 127.0.0.1:56437 #4 (4 connections now open)
2018-05-07T21:13:04.956+1200 I NETWORK [listener] connection accepted from 127.0.0.1:56438 #5 (5 connections now open)
2018-05-07T21:13:04.965+1200 I NETWORK [conn4] received client metadata from 127.0.0.1:56437 conn4: { driver: { name: "
nodejs", version: "3.0.2-1" }, os: { type: "Windows_NT", name: "win32", architecture: "ia32", version: "10.0.16299" }, p
latform: "Node.js v8.2.1, LE, mongodb-core: 3.0.2" }
2018-05-07T21:13:04.965+1200 I NETWORK [conn5] received client metadata from 127.0.0.1:56438 conn5: { driver: { name: "
nodejs", version: "3.0.2-1" }, os: { type: "Windows_NT", name: "win32", architecture: "ia32", version: "10.0.16299" }, p
latform: "Node.js v8.2.1, LE, mongodb-core: 3.0.2" }
2018-05-07T21:13:05.870+1200 I NETWORK [listener] connection accepted from 127.0.0.1:56439 #6 (6 connections now open)
2018-05-07T21:13:05.871+1200 I NETWORK [conn6] received client metadata from 127.0.0.1:56439 conn6: { driver: { name: "
nodejs", version: "3.0.2-1" }, os: { type: "Windows_NT", name: "win32", architecture: "ia32", version: "10.0.16299" }, p
latform: "Node.js v8.2.1, LE, mongodb-core: 3.0.2" }
2018-05-07T21:19:21.571+1200 I NETWORK [conn4] end connection 127.0.0.1:56437 (5 connections now open)
2018-05-07T21:19:25.138+1200 I NETWORK [conn5] end connection 127.0.0.1:56438 (4 connections now open)
2018-05-07T21:19:25.139+1200 I NETWORK [conn6] end connection 127.0.0.1:56439 (3 connections now open)

```

Figure 3: Mongod

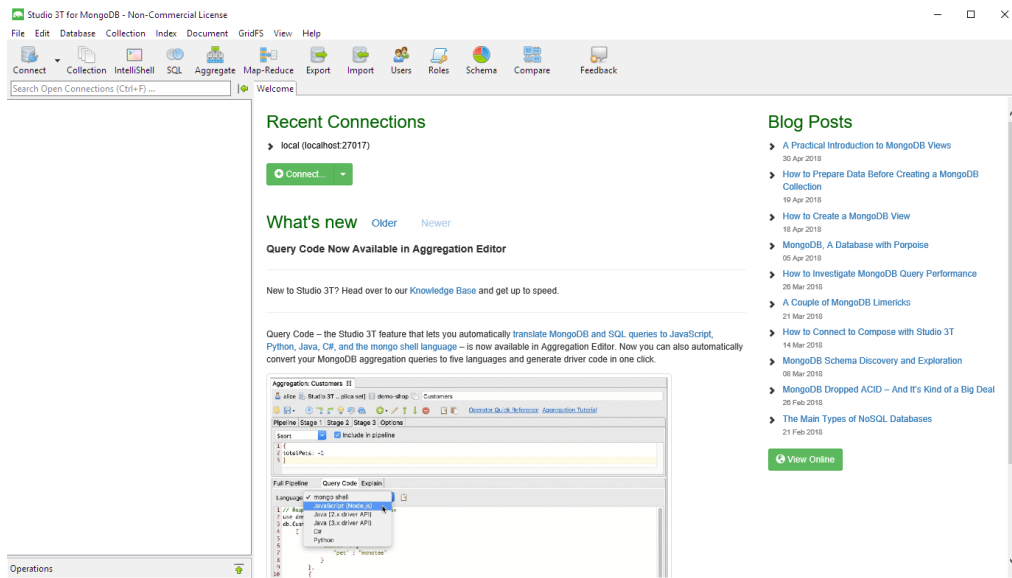


Figure 4: Studio 3T

Insert data

Code:

```
use admin
```

```
db.book.insert({  
  "_id" : "01",  
  "category" : "Novel",  
  "title" : "IT chapter 1",  
  "publisherID" : "PUB15010",  
  "author" : {authorFName: "Stephen", authorLName: "King"},  
  "published" : "17 July 2015",  
  "format" : "E-book",  
  "pages" : 345,  
  "price" : 25,  
  "comment" : [  
    "It freaks me out - LB",  
    "Interesting, twist plot"  
  ]  
})
```

```
use admin
```

```
db.book.insert({  
  "_id" : "02",  
  "category" : "Novel",  
  "title" : "System design and analysis",  
  "publisherID" : "PUB15010",  
  "author" : {authorFName: "Robert", authorLName: "Jackson"},  
  "published" : "27 Aug 2015",  
  "format" : "Hard cover",
```



```
"pages" : 560,  
"price" : 230,  
"comment" : [  
    "amazing"  
]  
})
```

use admin

```
db.book.insert({  
    "_id" : "03",  
    "category" : "Novel",  
    "title" : "Me before you",  
    "publisherID" : "PUB07018",  
    "author" : {authorFName: "Jojo", authorLName: "Moyes"},  
    "published" : "21 Jan 2016",  
    "format" : "a",  
    "pages" : 254,  
    "price" : 45,  
    "comment" : [  
        "Lovely and gentle, every word is the music note of the stunning song"  
    ]  
})
```

use admin

```
db.book.insert({  
    "_id" : "04",  
    "category" : "IT",  
    "title" : "Database for dummies",  
    "publisherID" : "PUB27062",  
    "author" : {authorFName: "Robert", authorLName: "Jackson"},  
    "published" : "05 Feb 2015",
```

```
"format" : "Hard cover",  
"pages" : 427,  
"price" : 154,  
"comment" : [  
    "Easy to learn and follow with simple words and basic level"  
]  
})
```

use admin

```
db.book.insert({  
    "_id" : "05",  
    "category" : "IT",  
    "title" : "Advance Java in industry world",  
    "publisherID" : "PUB27062",  
    "author" : {  
        "authorFName" : "Robert",  
        "authorLName" : "Jackson"  
    },  
    "published" : "22 Nov 2008",  
    "format" : "Hard cover",  
    "pages" : 369.0,  
    "price" : 178.0,  
    "comment" : [  
        "n/a"  
    ]  
})
```

Result:

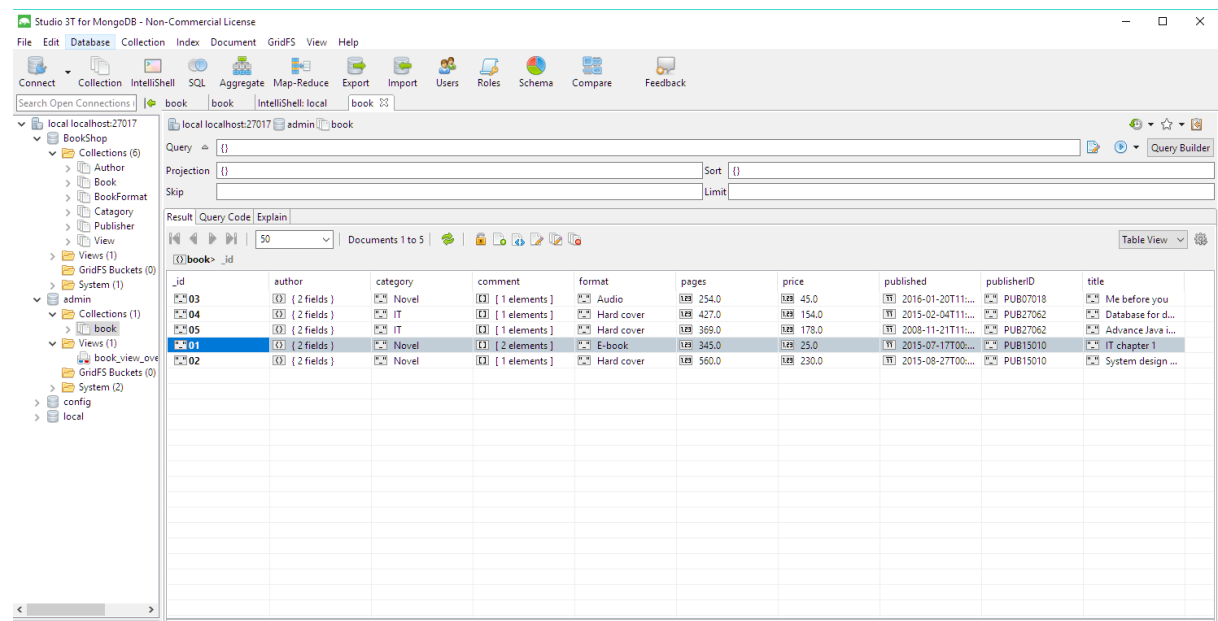


Figure 5: Database

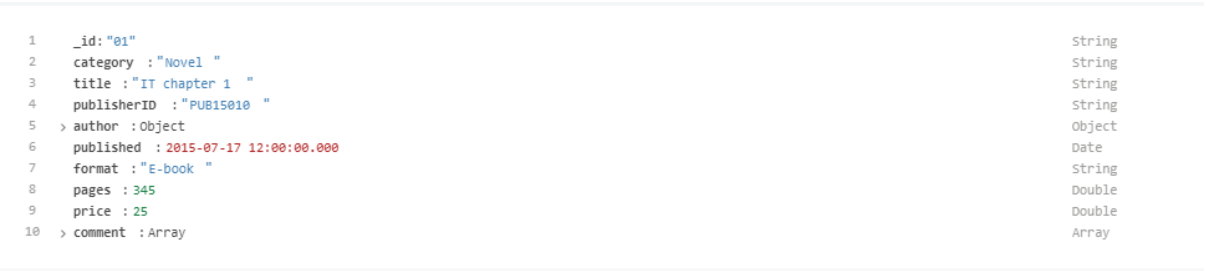


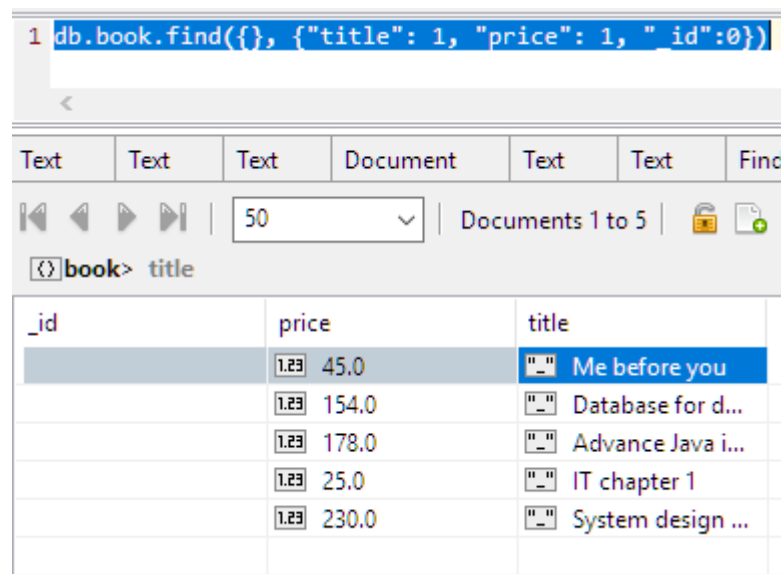
Figure 6: Data type

Query 1: List all the books Titles and their associated price

Code:

```
db.book.find({}, {title:1, price:1})
```

Result:



Text	Text	Text	Document	Text	Text	Find
50 Documents 1 to 5						
book> title						
_id	price	title				
	45.0	Me before you				
	154.0	Database for d...				
	178.0	Advance Java i...				
	25.0	IT chapter 1				
	230.0	System design ...				

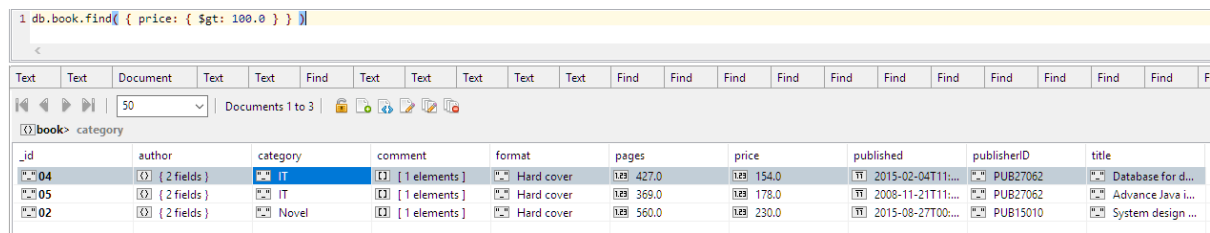
Figure 7: Query 1

Query 2: List all the information of the books which have a price higher than 100\$ and list their corresponding author name and surnames

Code:

```
db.book.find({}, {title:1, price:1})
```

Result:



_id	author	category	comment	format	pages	price	published	publisherID	title
04	{ 2 fields }	IT	[1 elements]	Hard cover	427.0	154.0	2015-02-04T11:...	PUB27062	Database for d...
05	{ 2 fields }	IT	[1 elements]	Hard cover	369.0	178.0	2008-11-21T11:...	PUB27062	Advance Java i...
02	{ 2 fields }	Novel	[1 elements]	Hard cover	560.0	230.0	2015-08-27T00:...	PUB15010	System design ...

Figure 8: Query 2

Query 3: Add a new book

Code:

use admin

```
db.book.insert({
  "_id" : "06",
  "category" : "Novel",
  "title" : "Explosion art",
  "publisherID" : "PUB15010",
  "author" : {
    "authorFName" : "Sana",
    "authorLName" : "Huynh"
  },
  "published" : "20 Mar 2004",
  "format" : "Hard cover",
  "pages" : 219.0,
  "price" : 172.0,
  "comment" : [
    "n/a"
  ]
})
```

Result:

_id	author	category	comment	format	pages	price	published	publisherID	title
03	{ 2 fields }	Novel	[1 elements]	Audio	254.0	45.0	2016-01-20T11:...	PUB07018	Me before you
04	{ 2 fields }	IT	[1 elements]	Hard cover	427.0	154.0	2015-02-04T11:...	PUB27062	Database for d...
05	{ 2 fields }	IT	[1 elements]	Hard cover	369.0	178.0	2008-11-21T11:...	PUB27062	Advance Java i...
01	{ 2 fields }	Novel	[2 elements]	E-book	345.0	25.0	2015-07-17T00:...	PUB15010	IT chapter 1
02	{ 2 fields }	Novel	[1 elements]	Hard cover	560.0	230.0	2015-08-27T00:...	PUB15010	System design ...
06	{ 2 fields }	Novel	[1 elements]	Hard cover	219.0	172.0	20 Mar 2004	PUB15010	Explosion art

Figure 9: Query 3

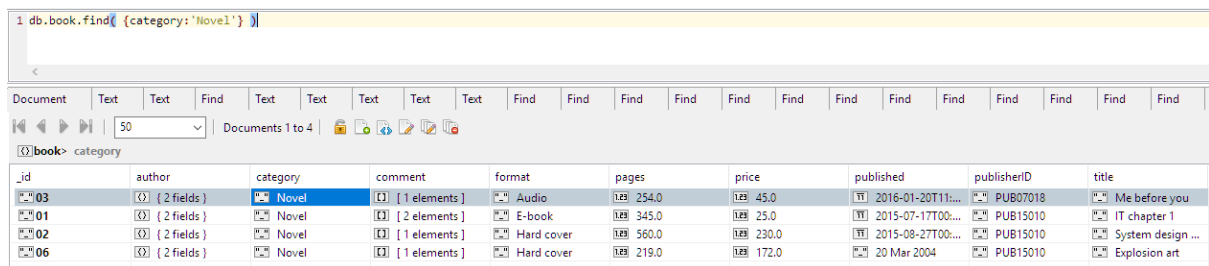
Query 4: List all the books which belong to a specific category

Code:

```
db.book.find( {category:'Novel'} )
```

(in this case I choose the category: “Novel”)

Result:



1 db.book.find({category:'Novel'})									
<									
Document	Text	Text	Find	Text	Text	Text	Text	Text	Text
50 Documents 1 to 4									
book-> category									
_id	author	category	comment	format	pages	price	published	publisherID	title
03	{ 2 fields }	Novel	[1 elements]	Audio	254.0	45.0	2016-01-20T11:...	PUB07018	Me before you
01	{ 2 fields }	Novel	[2 elements]	E-book	345.0	25.0	2015-07-17T00:...	PUB15010	IT chapter 1
02	{ 2 fields }	Novel	[1 elements]	Hard cover	560.0	230.0	2015-08-27T00:...	PUB15010	System design ...
06	{ 2 fields }	Novel	[1 elements]	Hard cover	219.0	172.0	20 Mar 2004	PUB15010	Explosion art

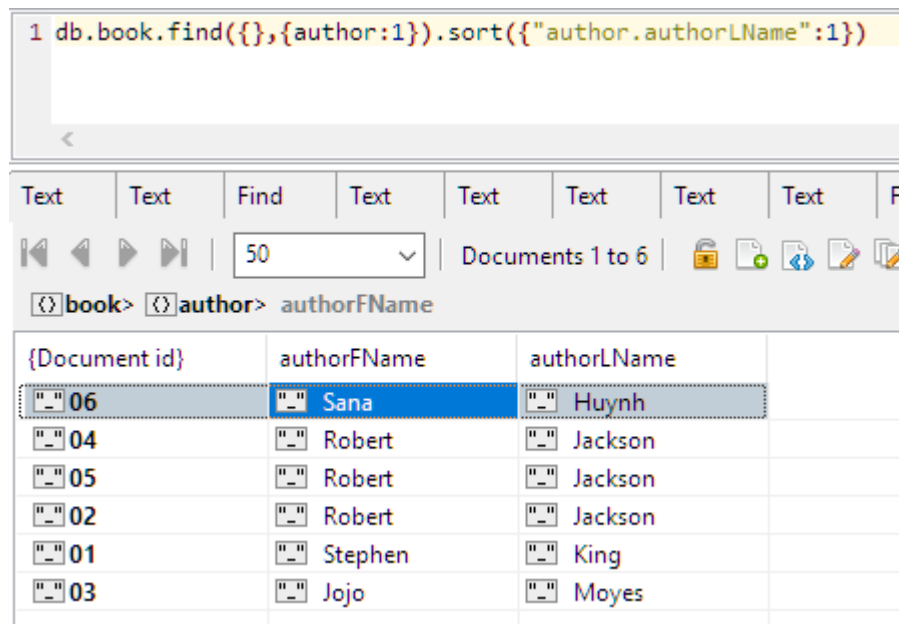
Figure 10: Query 4

Query 5: List all the author information ordered by their last name

Code:

```
db.book.find({}, {author:1}).sort({"author.authorLName":1})
```

Result:



{Document id}	authorFName	authorLName
"06	" Sana	" Huynh
"04	" Robert	" Jackson
"05	" Robert	" Jackson
"02	" Robert	" Jackson
"01	" Stephen	" King
"03	" Jojo	" Moyes

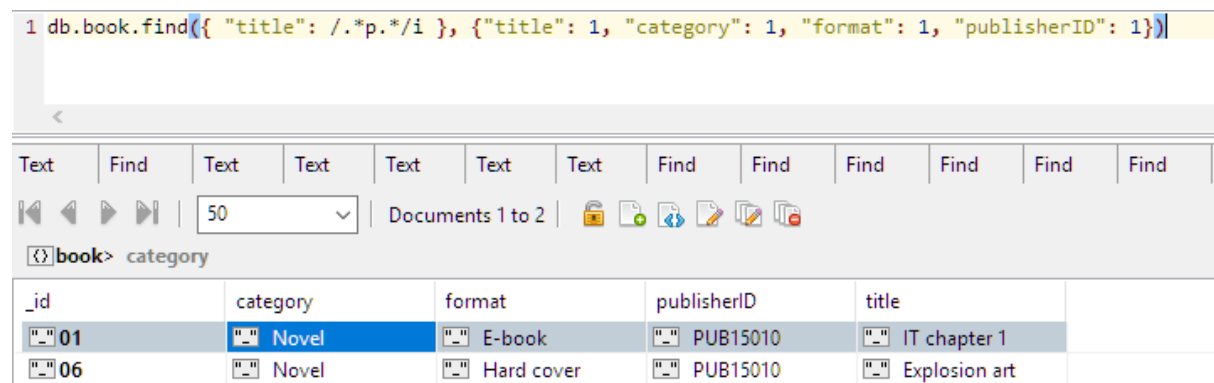
Figure 11: Query 5

Query 6: List the Title, Book Format (Description), Category (Description) and publisher name of all the books that contain the letter “P”

Code:

```
db.book.find({ "title": /. *p.*/i }, {"title": 1, "category": 1, "format": 1, "publisherID": 1})
```

Result:



The screenshot shows a MongoDB query interface. At the top, a code editor contains the query: `1 db.book.find({ "title": /. *p.*/i }, {"title": 1, "category": 1, "format": 1, "publisherID": 1})`. Below the editor is a toolbar with various icons and a dropdown menu set to 50. The main area displays the results of the query as a table with the following columns: `_id`, `category`, `format`, `publisherID`, and `title`. The results show two documents: one with `_id` 01, category Novel, format E-book, publisherID PUB15010, and title IT chapter 1; and another with `_id` 06, category Novel, format Hard cover, publisherID PUB15010, and title Explosion art.

<code>_id</code>	<code>category</code>	<code>format</code>	<code>publisherID</code>	<code>title</code>
01	Novel	E-book	PUB15010	IT chapter 1
06	Novel	Hard cover	PUB15010	Explosion art

Figure 12: Query 6

Query 7: Delete the book with the highest price

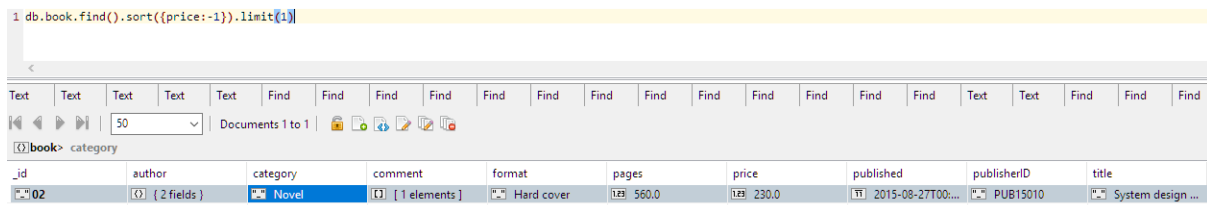
In this part, I do the pathway with 2 steps: finding the highest price then use that data (ID of book) to write other query to remove that document.

Code:

```
db.book.find().sort({price:-1}).limit(1)

try {
  db.book.deleteOne( { "_id" : "02" } );
} catch (e) {
  print(e);
}
```

Result:

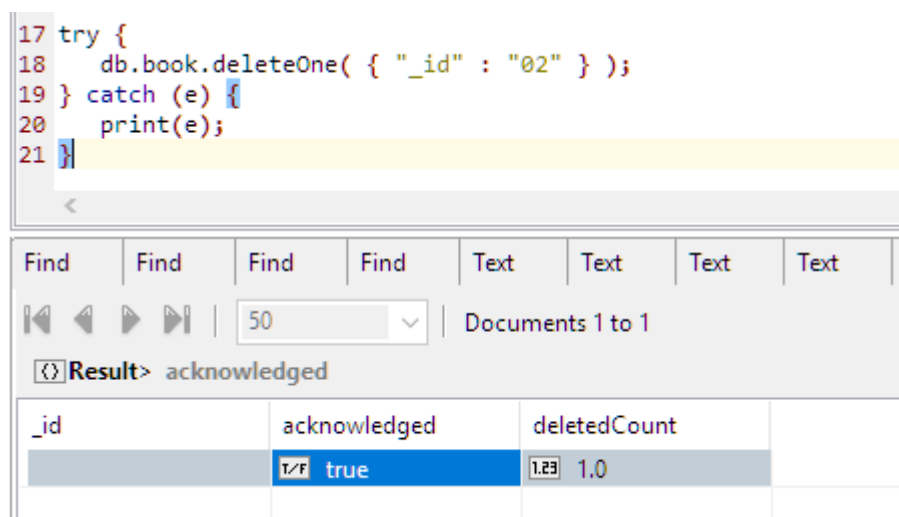


The screenshot shows the MongoDB Shell interface. The command `1 db.book.find().sort({price:-1}).limit(1)` is entered in the command line. Below the command line, a table displays the result of the query. The table has columns: `_id`, `author`, `category`, `comment`, `format`, `pages`, `price`, `published`, `publisherID`, and `title`. The first row shows the document with `_id` "02", which has the highest price of 230.0.

_id	author	category	comment	format	pages	price	published	publisherID	title
02	(2 fields)	Novel	[1 elements]	Hard cover	560.0	230.0	2015-08-27T00:...	PUB15010	System design ...

Figure 13: Query 7 - highest value

We can see that, in my collection, the book has the ID: 02 is the one has highest price (figure 13). Then, this ID is used as the filter for `deleteOne()` function (figure 14) and when check the collection again, we will see the book with ID 02 is deleted (figure 15).



The screenshot shows the MongoDB Shell interface. The code block contains the `deleteOne()` function call: `17 try { 18 db.book.deleteOne({ "_id" : "02" }); 19 } catch (e) { 20 print(e); 21 }`. Below the code, a table displays the result of the `deleteOne()` function. The table has columns: `_id`, `acknowledged`, and `deletedCount`. The first row shows the result for the deletion of the document with `_id` "02", where `acknowledged` is `true` and `deletedCount` is `1.0`.

_id	acknowledged	deletedCount
	true	1.0

Figure 14: deleteOne() function



















































_id	author	category	comment	format	pages	price	published	publisherD	title
 03	 { 2 fields }	 Novel	 [1 elements]	 Audio	 254.0	 45.0	 2016-01-20T11:...	 PUB07018	 Me before you
 04	 { 2 fields }	 IT	 [1 elements]	 Hard cover	 427.0	 154.0	 2015-02-04T11:...	 PUB27062	 Database for d...
 05	 { 2 fields }	 IT	 [1 elements]	 Hard cover	 369.0	 178.0	 2008-11-21T11:...	 PUB27062	 Advance Java i...
 01	 { 2 fields }	 Novel	 [2 elements]	 E-book	 345.0	 25.0	 2015-07-17T00:...	 PUB15010	 IT chapter 1
 06	 { 2 fields }	 Novel	 [1 elements]	 Hard cover	 219.0	 172.0	 20 Mar 2004	 PUB15010	 Explosion art

Figure 15: book collection after deleting highest price book