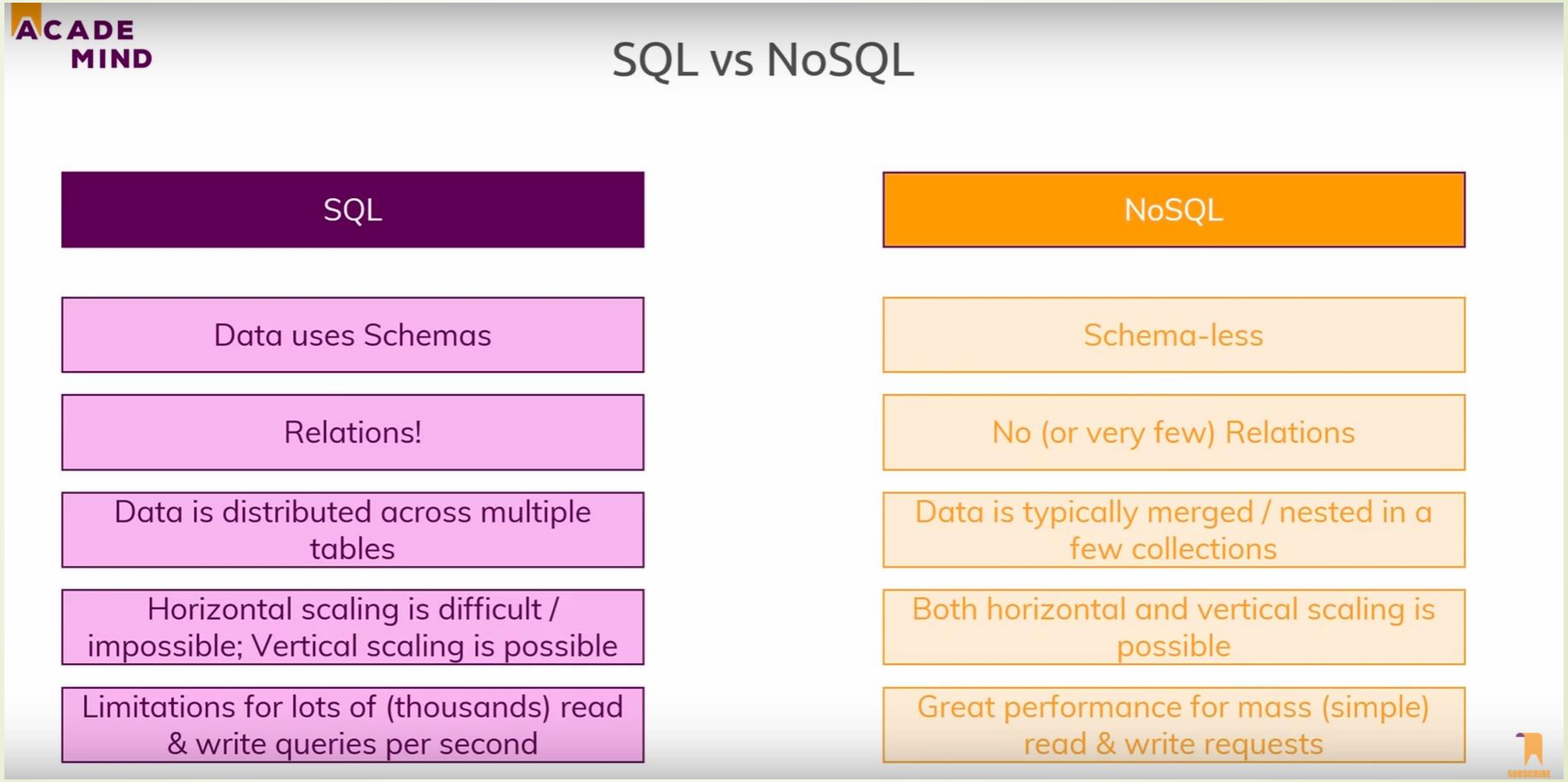
mongodb

SQL Vs. NoSQL



SQL vs NoSQL

SQL	NoSQL
Data uses Schemas	Schema-less
Relations!	No (or very few) Relations
Data is distributed across multiple tables	Data is typically merged / nested in a few collections
Horizontal scaling is difficult / impossible; Vertical scaling is possible	Both horizontal and vertical scaling is possible
Limitations for lots of (thousands) read & write queries per second	Great performance for mass (simple) read & write requests

Mongo-tools

- mongodump : is a utility for creating a binary export of the contents of a database
- mongorestore: loads data from either a binary database dump created by mongodump.
- mongoexport: is a utility that produces a JSON export of data stored in a MongoDB instance.
- mongoimport: is a tool imports content from an Extended JSON

Cursor

- A pointer to the result set of a query.
- Clients can iterate through a cursor to retrieve results.
- Cursor methods

```
var myCursor = db.users.find( { type: 2 } );

while (myCursor.hasNext()) {
   print(tojson(myCursor.next()));
   // your logic
}
```

Operators

- Query and Projection Operators (Read).
- Update Operators.
- Aggregation Operators.

Query and Projection Operators

Query Selectors

- Locate Data

Projection Operators

- Modify data presentation

Query Selectors

Comparison

- $eq
- $gt
- $gte
- $in
- $lt
- $lte
- $ne
- $nin

Logical

- $and
- $or
- $not
- $nor

Element

- $exist
- $type

Projection Operators

- $elemMatch
- $slice
- $

Update Operators

- $set
- $unset
- $inc
- $push
- $pullAll

Indexing

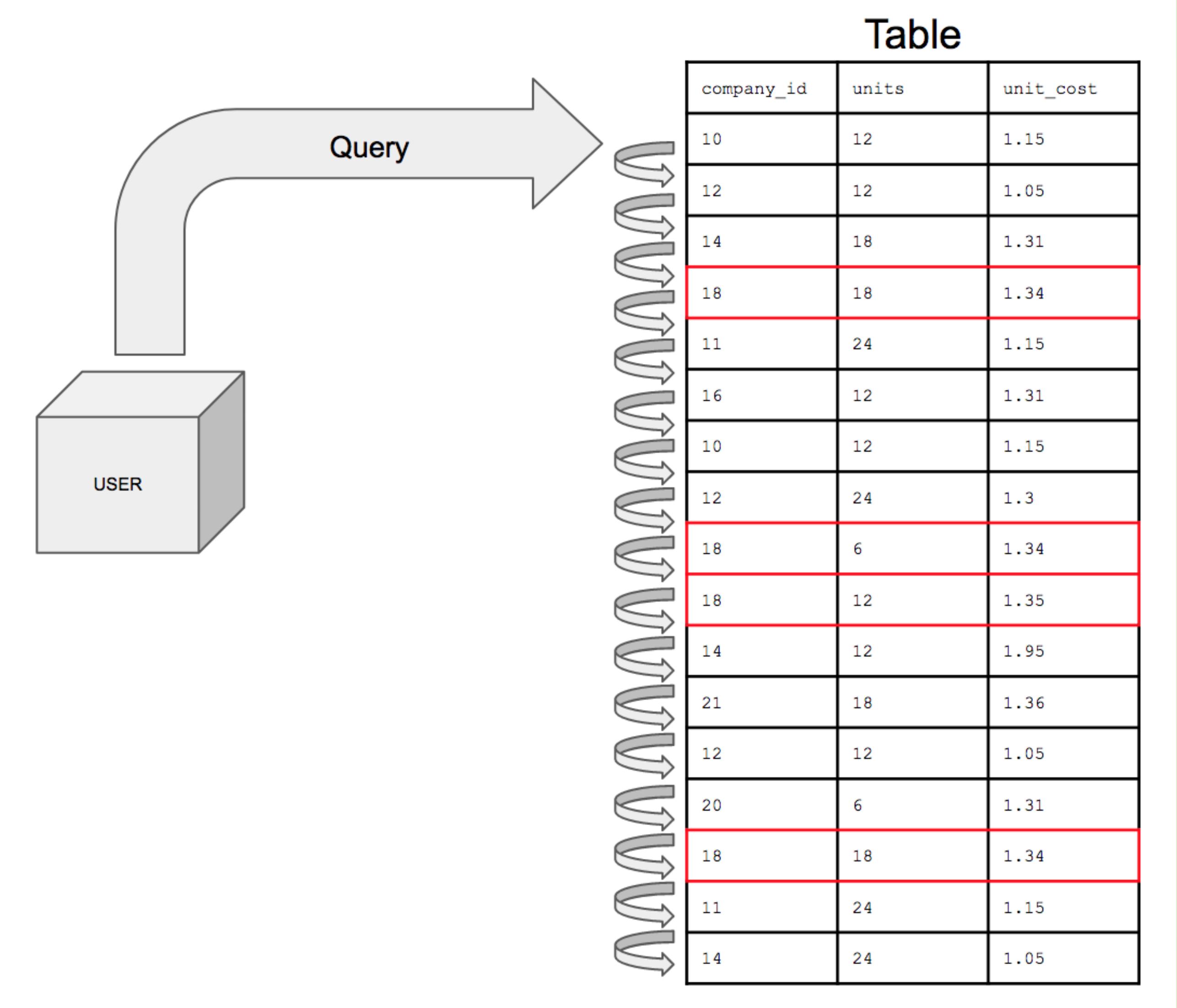
- “If you don’t find it in the index, look very carefully through the entire catalog.” – Sears, Roebuck, and Co., Consumers’ Guide 1897

The Working Set

- The working set is the portion of your data that's accessed frequently.
- If the working set fits in RAM you can serve most queries from the OS's in-memory cache, without waiting for the disk.

How find works

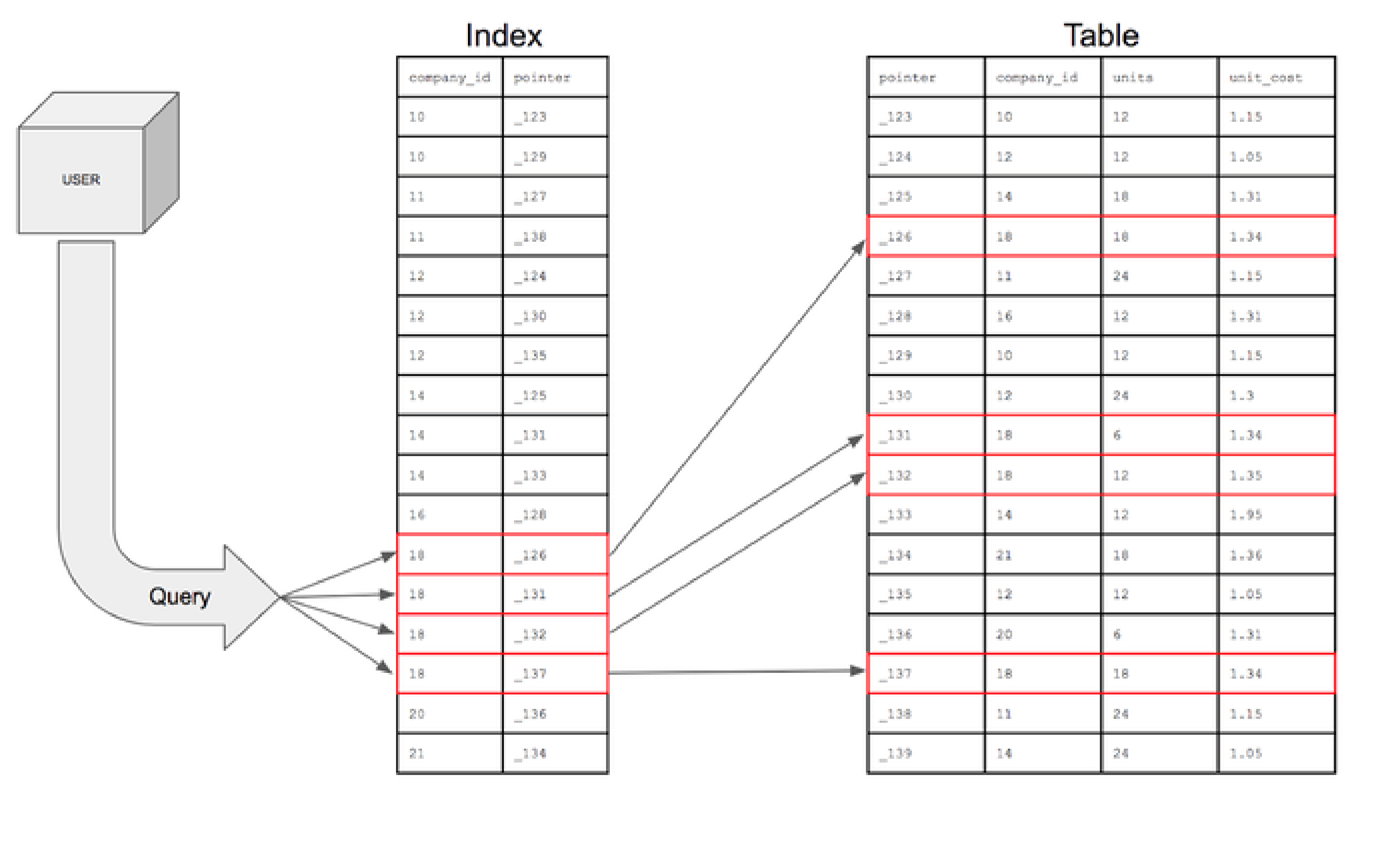
- Search the entire data to get the matched elements.



company_id	units	unit_cost
10	12	1.15
12	12	1.05
14	18	1.31
18	18	1.34
11	24	1.15
16	12	1.31
10	12	1.15
12	24	1.3
18	6	1.34
18	12	1.35
14	12	1.95
21	18	1.36
12	12	1.05
20	6	1.31
18	18	1.34
11	24	1.15
14	24	1.05

Indexing

- Search the index and then fetch the elements using pointers.



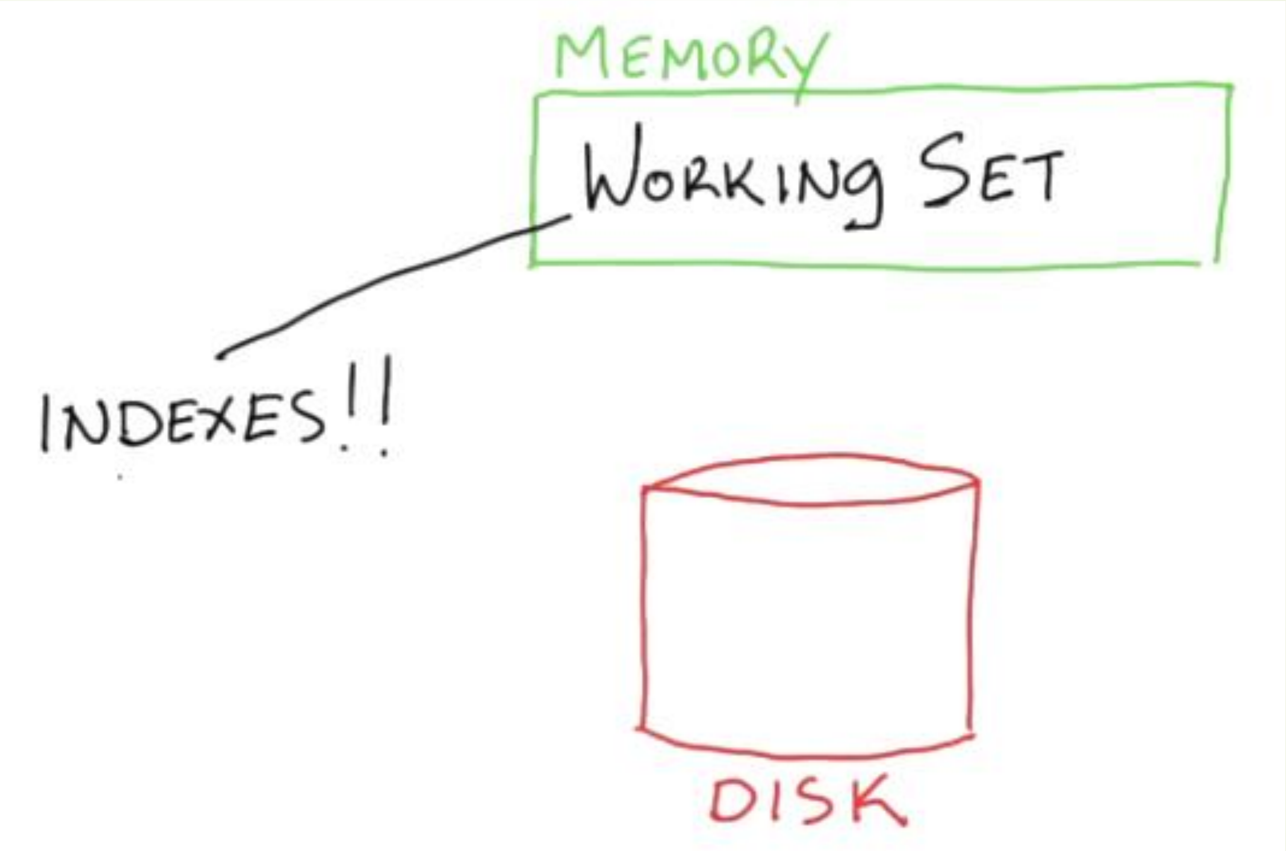
Index

company_id	pointer
10	_123
10	_129
11	_127
11	_138
12	_124
12	_130
12	_135
14	_125
14	_131
14	_133
16	_128
18	_126
18	_131
18	_132
18	_137
20	_136
21	_134

Table

pointer	company_id	units	unit_cost
_123	10	12	1.15
_124	12	12	1.05
_125	14	18	1.31
_126	18	18	1.34
_127	11	24	1.15
_128	16	12	1.31
_129	10	12	1.15
_130	12	24	1.3
_131	18	6	1.34
_132	18	12	1.35
_133	14	12	1.95
_134	21	18	1.36
_135	12	12	1.05
_136	20	6	1.31
_137	18	18	1.34
_138	11	24	1.15
_139	14	24	1.05

Indexing



Create an Index

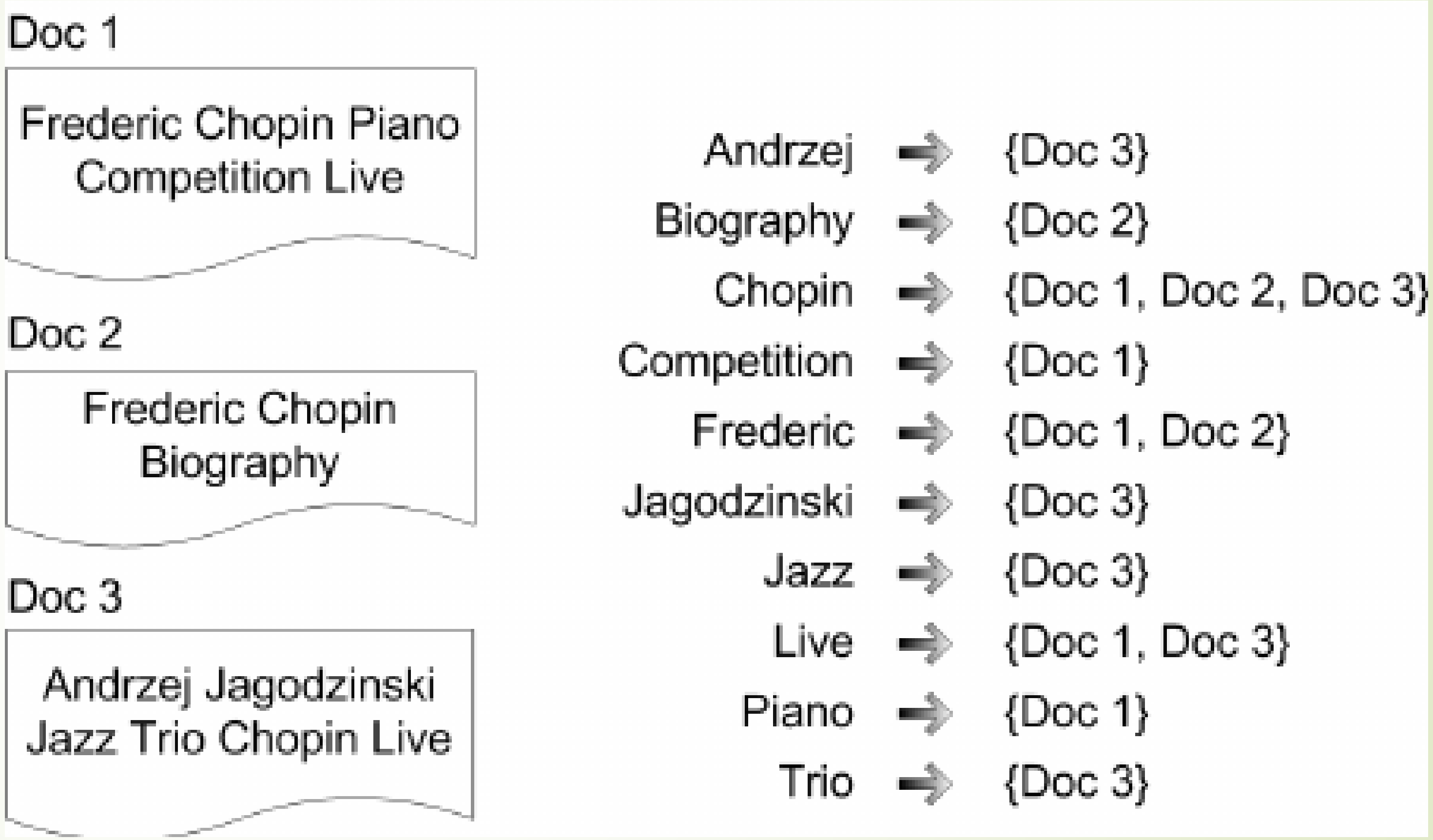
```
db.collection.createIndex( <key and index type specification>, <options> )

db.collection.createIndex( { name: -1 } )
```

Index Types

- Single Field
- Compound Index
- Multikey Index
- Text Indexes

Text Search

- Search Long Texts
- Remove Stop Words
- Stemming
- One Per Collection
- Cost Performance



Doc 1

Frederic Chopin Piano Competition Live

Doc 2

Frederic Chopin Biography

Doc 3

Andrzej Jagodzinski Jazz Trio Chopin Live

Andrzej	⇨	{Doc 3}
Biography	⇨	{Doc 2}
Chopin	⇨	{Doc 1, Doc 2, Doc 3}
Competition	⇨	{Doc 1}
Frederic	⇨	{Doc 1, Doc 2}
Jagodzinski	⇨	{Doc 3}
Jazz	⇨	{Doc 3}
Live	⇨	{Doc 1, Doc 3}
Piano	⇨	{Doc 1}
Trio	⇨	{Doc 3}

Lab 2

- Import Inventory Database using this command in terminal.

  mongorestore --db <database name> path_to_database
- select products with price greater than 1000 and less than 5000.
- display products which have phone number for vendor “using 2 ways ”
- display products which available in 4 stocks at the same time.
- increase all products by 500 EGP.
- replace stock #30 with #60 in all products.
- remove stock 70 from all products.
- display only product name and vendor phone number.
- display the most expensive product.
- Import books from json file into new database use **mongoimport.**
- Create text index on shortDescription field
- Retrieve all books which have“android”
- Retrieve all books which have "web applications" exactly