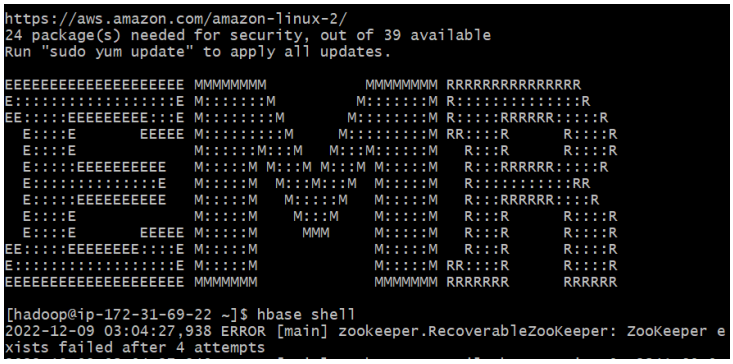
Girish Rajani
A20503736
CSP-554 Big Data Technologies
Homework 9

HBase:

Starting and entering the HBase shell



Exercise 1) (1 point)
Create an HBase table and then use describe command

```
=> Hbase::Table - csp554Tbl
hbase(main):002:0> describe 'csp554Tbl'
Table csp554Tbl is ENABLED
csp554Tbl
COLUMN FAMILIES DESCRIPTION
{NAME => 'cf1', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEE
P_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COM
PRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '655
36', REPLICATION_SCOPE => '0'}
{NAME => 'cf2', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEE
P_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COM
PRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '655
36', REPLICATION_SCOPE => '0'}
2 row(s) in 0.0410 seconds
```

Exercise 2) (1 point)
Put data in the table created from the previous exercise

```
hbase(main):004:0> put 'csp554Tbl','Row2','cf1:name','Ahmed'
0 row(s) in 0.0060 seconds

hbase(main):005:0> put 'csp554Tbl','Row1','cf2:job','Pilot'
0 row(s) in 0.0220 seconds

hbase(main):006:0> put 'csp554Tbl','Row2','cf2:job','Doctor'
0 row(s) in 0.0090 seconds

hbase(main):007:0> put 'csp554Tbl','Row1','cf2:level','LZ3'
0 row(s) in 0.0080 seconds

hbase(main):008:0> put 'csp554Tbl','Row2','cf2:level','AR7'
0 row(s) in 0.0100 seconds
```

Execute SCAN command to return all rows, column families and columns on the table

```
hbase(main):010:0> scan 'csp554Tbl'
ROW                   COLUMN+CELL
 Row1                 column=cf1:name, timestamp=1670557177933, value=Sam
 Row1                 column=cf2:job, timestamp=1670557246110, value=Pilot
 Row1                 column=cf2:level, timestamp=1670557261333, value=LZ3
 Row2                 column=cf1:name, timestamp=1670557190385, value=Ahmed
 Row2                 column=cf2:job, timestamp=1670557253909, value=Doctor
 Row2                 column=cf2:level, timestamp=1670557272288, value=AR7
2 row(s) in 0.0260 seconds
```

Exercise 3) (1 point)

```
hbase(main):011:0> get 'csp554Tbl', 'Row1', 'cf2:level'
COLUMN                CELL
 cf2:level            timestamp=1670557261333, value=LZ3
1 row(s) in 0.0230 seconds
```

Showing command used to get the value associated with row (Row1), column family (cf2) and column,qualifier (level)

Command used: get 'csp554Tbl', 'Row1', 'cf2:level'

Exercise 4) (1 point)

```
hbase(main):012:0> get 'csp554Tbl', 'Row2', 'cf1:name'
COLUMN                CELL
 cf1:name             timestamp=1670557190385, value=Ahmed
1 row(s) in 0.0140 seconds
```

Showing command used to get the value associated with row (Row2), column family (cf1) and column/qualifier (name)

Command used: get 'csp554Tbl', 'Row2', 'cf1:name'

Exercise 5) (1 point)

```
hbase(main):021:0> scan 'csp554Tbl',{LIMIT=>2}
ROW                   COLUMN+CELL
 Row1                 column=cf1:name, timestamp=1670557177933, value=Sam
 Row1                 column=cf2:job, timestamp=1670557246110, value=Pilot
 Row1                 column=cf2:level, timestamp=1670557261333, value=LZ3
 Row2                 column=cf1:name, timestamp=1670557190385, value=Ahmed
 Row2                 column=cf2:job, timestamp=1670557253909, value=Doctor
 Row2                 column=cf2:level, timestamp=1670557272288, value=AR7
2 row(s) in 0.0080 seconds
```

Showing SCAN command used to return information about only two rows using the LIMIT modifier

Command used: scan 'csp554Tbl',{LIMIT=>2}

Cassandra

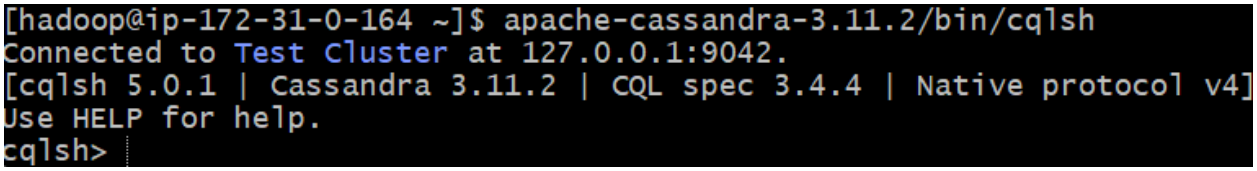
Exercise 1) (1 point)

Apache Cassandra is a top contending online transaction processing and scalable distributed database that enables the processing of huge datasets which are managed through thousands of cluster nodes across multiple data centers. Cassandra can be used throughout various applications such as the Internet of Things, social networking, fraud detection, and much more, which, in my opinion, makes it a great option when working with time series data while also offering scalability and fault tolerance. Cassandra incorporates a very easy-to-understand language called CQL (Cassandra Query Language) that allows for high read/write access and handling of failures. Based on this paper and personal experience, Cassandra has a lot to offer, and its popularity is no surprise due to its speed, flexibility, scalability, data replication, and so much more. However, it cannot go without mentioning that Apache Cassandra has some challenges, one of the biggest being the data modeling methodology for Cassandra.

Traditional data modeling methods cannot be applied to Cassandra due to its complexity and how the technology works, especially when working with queries. This paper introduced the first query-driven big data modeling methodology for Apache Cassandra that I believe alleviates such a challenge. This model showed a different approach in the schema design, data nesting, and data duplication aspects compared to the traditional data modeling methods, such as those seen in relational databases. A novel visualization technique called Chebotko Diagrams was introduced, which captures complex logical and physical data models. A powerful data modeling tool called KDM was introduced, automating Cassandra's many complex data modeling tasks. I strongly believe that all of these features included in the proposed data modeling methodology for Apache Cassandra make it a very powerful and practical tool for managing large data sets within a distributed database system.

Exercise 2) (1 point)

Step C



Showing command used in the Cqlsh-Term terminal to start the command line interface csqlsh

Step D

```
giris@LAPTOP-18TD7ESB MINGW64 ~
$ scp -i d:/users/giris/downloads/emr-key-pair.pem d:/users/giris/downloads/init
.cql hadoop@ec2-44-203-240-149.compute-1.amazonaws.com:/home/hadoop
init.cql                                      100%    0     0.0KB/s   00:00
```

Using scp command to upload init.cql file to the EMR master node that was created locally

```
cqlsh> source'./init.cql';
cqlsh> describe keyspaces;

system_schema  system_auth  system  system_distributed  a20503736  system_traces

cqlsh> USE A20503736;
```

b) and c) executing the init.cql file and checking if the file script has created a keyspace

d) Making the keyspace the default by entering the following in the CQL shell.

```
giris@LAPTOP-18TD7ESB MINGW64 ~
$ scp -i d:/users/giris/downloads/emr-key-pair.pem d:/users/giris/downloads/ex2.
cql hadoop@ec2-44-203-240-149.compute-1.amazonaws.com:/home/hadoop
ex2.cql                                       100%  123     6.2KB/s   00:00
```

Using scp command to upload the ex2.cql file which will create a table named Music

```
cqlsh:a20503736> source'./ex2.cql';
cqlsh:a20503736> describe table Music;

CREATE TABLE a20503736.music (
    artistname text,
    albumname text,
    cost int,
    numbersold int,
    PRIMARY KEY (artistname, albumname)
) WITH CLUSTERING ORDER BY (albumname ASC)
    AND bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCom
pactionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandr
a.io.compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair_chance = 0.0
    AND speculative_retry = '99PERCENTILE';
```

Showing output of describe table Music after executing ex2.cql in the CQL shell

Exercise 3) (1 point)

```
giris@LAPTOP-18TD7ESB MINGW64 ~
$ scp -i d:/users/giris/downloads/emr-key-pair.pem d:/users/giris/downloads/ex3.
cql hadoop@ec2-44-203-240-149.compute-1.amazonaws.com:/home/hadoop
ex3.cql                                       100%  274    12.4KB/s   00:00
```

```
hadoop@ip-172-31-0-164:~
INSERT INTO Music (artistName, albumName, numberSold, Cost) VALUES ('Mozart','Greatest Hits',100000, 10);
INSERT INTO Music (artistName, albumName, numberSold, Cost) VALUES ('Taylor Swift','Fearless',2300000, 15);
INSERT INTO Music (artistName, albumName, numberSold, Cost) VALUES ('Black Sabbath','Paranoid',534000, 12);
INSERT INTO Music (artistName, albumName, numberSold, Cost) VALUES ('Katy Perry','Prism',800000, 16);
INSERT INTO Music (artistName, albumName, numberSold, Cost) VALUES ('Katy Perry','Teenage Dream',750000, 14);
~
```

a) Showing content of ex3.cql

```
cqlsh:a20503736> source'./ex3.cql';
cqlsh:a20503736> select * from Music;

 artistname    | albumname     | cost | numbersold
---------------+---------------+------+------------
        Mozart | Greatest Hits |   10 |     100000
 Black Sabbath |      Paranoid |   12 |     534000
  Taylor Swift |      Fearless |   15 |    2300000
    Katy Perry |         Prism |   16 |     800000
    Katy Perry | Teenage Dream |   14 |     750000

(5 rows)
cqlsh:a20503736>
```

b) Showing output of 'SELECT * FROM Music;' command

Exercise 4) (1 point)

```
giris@LAPTOP-18TD7ESB MINGW64 ~
$ scp -i d:/users/giris/downloads/emr-key-pair.pem d:/users/giris/downloads/ex4.
cql hadoop@ec2-44-203-240-149.compute-1.amazonaws.com:/home/hadoop
ex4.cql                                       100%   52     2.4KB/s   00:00
```

```
hadoop@ip-172-31-0-164:~
SELECT * FROM Music WHERE artistName = 'Katy Perry';
~
```

a) Showing content of ex4.cql

b)

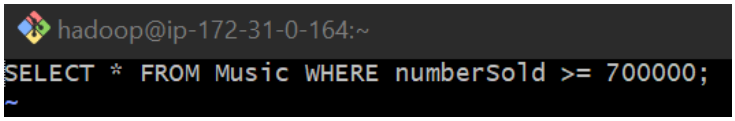
```
cqlsh:a20503736> source'./ex4.cql';

 artistname | albumname     | cost | numbersold
------------+---------------+------+------------
 Katy Perry |         Prism |   16 |     800000
 Katy Perry | Teenage Dream |   14 |     750000
```

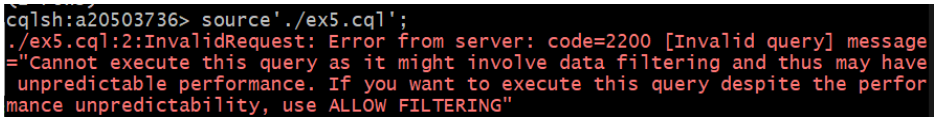
c) Showing output after executing ex4.cql

Exercise 5) (1 point)

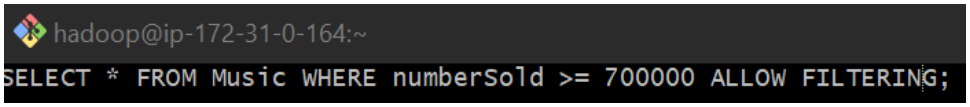
```
giris@LAPTOP-18TD7ESB MINGW64 ~
$ scp -i d:/users/giris/downloads/emr-key-pair.pem d:/users/giris/downloads/ex5.
cql hadoop@ec2-44-203-240-149.compute-1.amazonaws.com:/home/hadoop
ex5.cql                                       100%   47     2.2KB/s   00:00
```



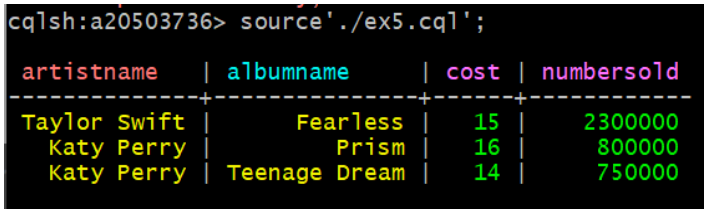
Showing initial content of ex5.cql



Showing error received after executing ex5.cql



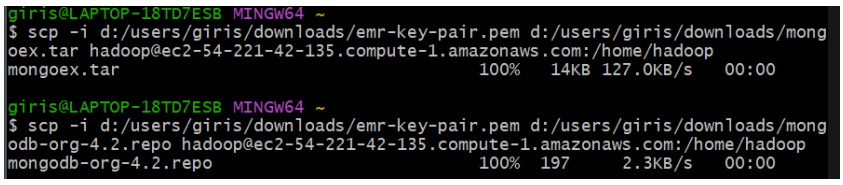
To fix the error, we use ALLOW FILTERING to the content of ex5.cql



Showing output after executing ex5.cql

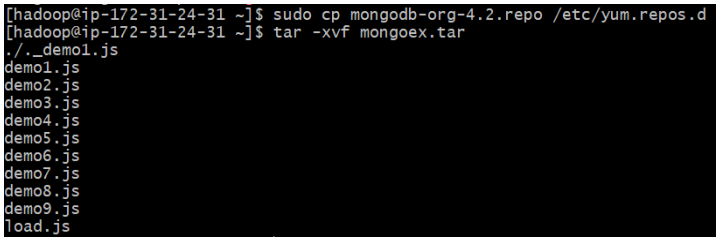
MongoDB

Step B



Scp both mongoex.tar and mongodb-org-4.2.repo to the master node

Step C - Install assignment software



sudo yum install -y mongodb-org-4.2.15 mongodb-org-server-4.2.15 mongodb-org-shell4.2.15 mongodb-org-mongos-4.2.15 mongodb-org-tools-4.2.15

Step D - Install and start MongoDB

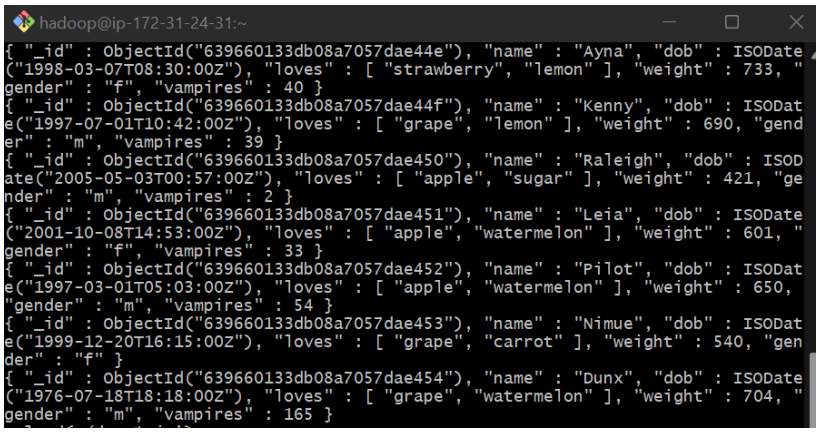
```
[hadoop@ip-172-31-24-31 ~]$ sudo yum install -y mongodb-org-4.2.15 mongodb-org-s
erver-4.2.15 mongodb-org-shell4.2.15 mongodb-org-mongos-4.2.15mongodb-org-tools-
4.2.15
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                                | 3.7 kB     00:00
mongodb-org-4.2                                           | 2.6 kB     00:00
mongodb-org-4.2/primary_db                                  | 107 kB   00:00
3 packages excluded due to repository priority protections
```

Start mongodb:

```
Complete!
[hadoop@ip-172-31-24-31 ~]$ sudo systemctl start mongod
```

Step G

Showing output after executing ‘db.unicorns.find();’ command to confirm all works well



Exercise 1) (1 point)

```
> db.unicorns.find( { weight: { $lt: 500 } } )
{ "_id" : ObjectId("639660133db08a7057dae44a"), "name" : "Aurora", "dob" : ISODa
te("1991-01-24T13:00:00Z"), "loves" : [ "carrot", "grape" ], "weight" : 450, "ge
nder" : "f", "vampires" : 43 }
{ "_id" : ObjectId("639660133db08a7057dae450"), "name" : "Raleigh", "dob" : ISOD
ate("2005-05-03T00:57:00Z"), "loves" : [ "apple", "sugar" ], "weight" : 421, "ge
nder" : "m", "vampires" : 2 }
```

Showing all unicorns having weight less than 500 pounds

Command used:
db.unicorns.find({ weight: { $lt: 500 } })

Exercise 2) (1 point)

```
> db.unicorns.find( { loves: "apple" } )
{ "_id" : ObjectId("639660133db08a7057dae44c"), "name" : "Roooooodles", "dob" :
ISODate("1979-08-18T18:44:00Z"), "loves" : [ "apple" ], "weight" : 575, "gender"
 : "m", "vampires" : 99 }
{ "_id" : ObjectId("639660133db08a7057dae44d"), "name" : "Solnara", "dob" : ISOD
ate("1985-07-04T02:01:00Z"), "loves" : [ "apple", "carrot", "chocolate" ], "weig
ht" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("639660133db08a7057dae450"), "name" : "Raleigh", "dob" : ISOD
ate("2005-05-03T00:57:00Z"), "loves" : [ "apple", "sugar" ], "weight" : 421, "ge
nder" : "m", "vampires" : 2 }
{ "_id" : ObjectId("639660133db08a7057dae451"), "name" : "Leia", "dob" : ISODate
("2001-10-08T14:53:00Z"), "loves" : [ "apple", "watermelon" ], "weight" : 601, "
gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("639660133db08a7057dae452"), "name" : "Pilot", "dob" : ISODat
e("1997-03-01T05:03:00Z"), "loves" : [ "apple", "watermelon" ], "weight" : 650,
"gender" : "m", "vampires" : 54 }
```

Showing all unicorns who loves apples

Command used:
db.unicorns.find({ loves: "apple" })

Exercise 3) (1 point)

```
> db.unicorns.insertOne( {name: "Malini", dob: "2008-03-11", loves: ["pears","gr
apes"], weight: 450, gender: "F", vampires: 23, horns: 1})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("6396678e3db08a7057dae455")
}
```

Showing command and output after adding a new unicorn with the given information
Command used:
db.unicorns.insertOne({name: "Malini", dob: "2008-03-11", loves: ["pears","grapes"], weight: 450, gender: "F", vampires: 23, horns: 1})

```
> db.unicorns.find();
{ "_id" : ObjectId("639660133db08a7057dae449"), "name" : "Horny", "dob" : ISODat
e("1992-03-13T07:47:00Z"), "loves" : [ "carrot", "papaya" ], "weight" : 600, "ge
nder" : "m", "vampires" : 63 }
{ "_id" : ObjectId("639660133db08a7057dae44a"), "name" : "Aurora", "dob" : ISODa
te("1991-01-24T13:00:00Z"), "loves" : [ "carrot", "grape" ], "weight" : 450, "ge
nder" : "f", "vampires" : 43 }
{ "_id" : ObjectId("639660133db08a7057dae44b"), "name" : "Unicrom", "dob" : ISOD
ate("1973-02-09T22:10:00Z"), "loves" : [ "energon", "redbull" ], "weight" : 984,
 "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("639660133db08a7057dae44c"), "name" : "Roooooodles", "dob" :
ISODate("1979-08-18T18:44:00Z"), "loves" : [ "apple" ], "weight" : 575, "gender"
 : "m", "vampires" : 99 }
{ "_id" : ObjectId("639660133db08a7057dae44d"), "name" : "Solnara", "dob" : ISOD
ate("1985-07-04T02:01:00Z"), "loves" : [ "apple", "carrot", "chocolate" ], "weig
ht" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("639660133db08a7057dae44e"), "name" : "Ayna", "dob" : ISODate
("1998-03-07T08:30:00Z"), "loves" : [ "strawberry", "lemon" ], "weight" : 733, "
gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("639660133db08a7057dae44f"), "name" : "Kenny", "dob" : ISODat
e("1997-07-01T10:42:00Z"), "loves" : [ "grape", "lemon" ], "weight" : 690, "gend
er" : "m", "vampires" : 39 }
{ "_id" : ObjectId("639660133db08a7057dae450"), "name" : "Raleigh", "dob" : ISOD
ate("2005-05-03T00:57:00Z"), "loves" : [ "apple", "sugar" ], "weight" : 421, "ge
nder" : "m", "vampires" : 2 }
{ "_id" : ObjectId("639660133db08a7057dae451"), "name" : "Leia", "dob" : ISODate
("2001-10-08T14:53:00Z"), "loves" : [ "apple", "watermelon" ], "weight" : 601, "
gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("639660133db08a7057dae452"), "name" : "Pilot", "dob" : ISODat
e("1997-03-01T05:03:00Z"), "loves" : [ "apple", "watermelon" ], "weight" : 650,
"gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("639660133db08a7057dae453"), "name" : "Nimue", "dob" : ISODat
e("1999-12-20T16:15:00Z"), "loves" : [ "grape", "carrot" ], "weight" : 540, "gen
der" : "f" }
{ "_id" : ObjectId("639660133db08a7057dae454"), "name" : "Dunx", "dob" : ISODate
("1976-07-18T18:18:00Z"), "loves" : [ "grape", "watermelon" ], "weight" : 704, "
gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("6396678e3db08a7057dae455"), "name" : "Malini", "dob" : "2008
-03-11", "loves" : [ "pears", "grapes" ], "weight" : 450, "gender" : "F", "vampi
res" : 23, "horns" : 1 }
>
```

Using db.unicorns.find() to confirm that the new unicorn Malini has been added successfully

Exercise 4) (1 point)

```
> db.unicorns.updateOne({name:"Malini"}, {$set:{loves:["pears","grapes","apricots"]}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
>
```

Showing command used to update the Malinin record to add apricots to the list of things Malini loves

Command used:
db.unicorns.updateOne({name:"Malini"}, {$set:{loves:["pears","grapes","apricots"]}})

```
> db.unicorns.find();
{ "_id" : ObjectId("639660133db08a7057dae449"), "name" : "Horny", "dob" : ISODat
e("1992-03-13T07:47:00Z"), "loves" : [ "carrot", "papaya" ], "weight" : 600, "ge
nder" : "m", "vampires" : 63 }
{ "_id" : ObjectId("639660133db08a7057dae44a"), "name" : "Aurora", "dob" : ISODa
te("1991-01-24T13:00:00Z"), "loves" : [ "carrot", "grape" ], "weight" : 450, "ge
nder" : "f", "vampires" : 43 }
{ "_id" : ObjectId("639660133db08a7057dae44b"), "name" : "Unicrom", "dob" : ISOD
ate("1973-02-09T22:10:00Z"), "loves" : [ "energon", "redbull" ], "weight" : 984,
 "gender" : "m", "vampires" : 182 }
{ "_id" : ObjectId("639660133db08a7057dae44c"), "name" : "Roooooodles", "dob" :
ISODate("1979-08-18T18:44:00Z"), "loves" : [ "apple" ], "weight" : 575, "gender"
 : "m", "vampires" : 99 }
{ "_id" : ObjectId("639660133db08a7057dae44d"), "name" : "Solnara", "dob" : ISOD
ate("1985-07-04T02:01:00Z"), "loves" : [ "apple", "carrot", "chocolate" ], "weig
ht" : 550, "gender" : "f", "vampires" : 80 }
{ "_id" : ObjectId("639660133db08a7057dae44e"), "name" : "Ayna", "dob" : ISODate
("1998-03-07T08:30:00Z"), "loves" : [ "strawberry", "lemon" ], "weight" : 733, "
gender" : "f", "vampires" : 40 }
{ "_id" : ObjectId("639660133db08a7057dae44f"), "name" : "Kenny", "dob" : ISODat
e("1997-07-01T10:42:00Z"), "loves" : [ "grape", "lemon" ], "weight" : 690, "gend
er" : "m", "vampires" : 39 }
{ "_id" : ObjectId("639660133db08a7057dae450"), "name" : "Raleigh", "dob" : ISOD
ate("2005-05-03T00:57:00Z"), "loves" : [ "apple", "sugar" ], "weight" : 421, "ge
nder" : "m", "vampires" : 2 }
{ "_id" : ObjectId("639660133db08a7057dae451"), "name" : "Leia", "dob" : ISODate
("2001-10-08T14:53:00Z"), "loves" : [ "apple", "watermelon" ], "weight" : 601, "
gender" : "f", "vampires" : 33 }
{ "_id" : ObjectId("639660133db08a7057dae452"), "name" : "Pilot", "dob" : ISODat
e("1997-03-01T05:03:00Z"), "loves" : [ "apple", "watermelon" ], "weight" : 650,
"gender" : "m", "vampires" : 54 }
{ "_id" : ObjectId("639660133db08a7057dae453"), "name" : "Nimue", "dob" : ISODat
e("1999-12-20T16:15:00Z"), "loves" : [ "grape", "carrot" ], "weight" : 540, "gen
der" : "f" }
{ "_id" : ObjectId("639660133db08a7057dae454"), "name" : "Dunx", "dob" : ISODate
("1976-07-18T18:18:00Z"), "loves" : [ "grape", "watermelon" ], "weight" : 704, "
gender" : "m", "vampires" : 165 }
{ "_id" : ObjectId("6396678e3db08a7057dae455"), "name" : "Malini", "dob" : "2008
-03-11", "loves" : [ "pears", "grapes", "apricots" ], "weight" : 450, "gender" :
 "F", "vampires" : 23, "horns" : 1 }
>
```

Showing output of the db.unicorns.find() command to confirm that Malini's record has been updated and apricots has been added to the list of things Malini loves

Exercise 5) (1 point)

```
> db.unicorns.deleteMany( {weight: {$gt:600} } )
{ "acknowledged" : true, "deletedCount" : 6 }
```

Showing code used to delete all unicorns with weight more than 600 pounds

Command used:
db.unicorns.deleteMany({weight: {$gt:600} })

Showing output of db.unicorns.find() command to confirm that all unicorns with weight more than 600 pounds have been deleted successfully