data Integration and processing.

Week 2

video 1

Querying JSON with mangoDB :-

JSON ⇒ semi-structured data

- mango DB is a collection of documet.

key-value pair ⇒ atomic elemet.

value ⇒ array; An array is a list.
Query operations can either be on its position on it either be on position in the list / value.

Top level array doesn't have a key, by default it is called db.

key value peers are structured as tuples.
db.collection.find (<queryfilter>, <projection>).<cursor modifier>

Collection :- which document collection to use similar to from clause.

if the name of the collection is beers,
⇒ db.beeers.find.

query filter :- lists all cod^n's that the retrieved document should satisfy,
⇒ where clause.

projection class :- list of variables that we want to see in the o/p

NOTES

cursor modifier :-
The word cursor relates back to SQL where cursor is defined as a block of results that is returned to the user in one chunk

query 1 : Wants everything from Beers .

SQL :- SELECT *
FROM Beers

mangodB :- db beers.find ()

query 2 :- SQL :- SELECT beer, price
FROM sells

12:00 MangoDB :- db.sells.find (
{ },
{ beer : 1, price : 1 } [{ beer : 1,
price : 1,
_id : 0 }]
)

2:00

Query 3 :- SQL :- SELECT manf
3:00 FROM Beers
WHERE name = 'Heineken'

4:00 MangoDB :- db.beers.find (
5:00 { name : "Heineken" }, { manf : 1, _id : 0 }
)

6:00

Query 4 :- SQL :- SELECT DISTINCT beer, price
7:00 FROM sells
WHERE price > 15

MangoDB :- db. sells. distinct (
NOTES
{ price : { $gt : 15 } },
{ beer : 1, price : 1, _id : 0 }
)

Some Operations of Mango DB :-

$eq → Matches values that are equal to a specific value.

$gt → Matches values that are greater than a specified value.

$gte → —||— that are greater than or equal to a specified value.

$lt → —||— that are less than a specified value.

$lte → —||— that are less than or equal to a specified value.

$ne → —||— not equal to a specified value.

$in → ~~—||—~~ Matches any of the values specified in the array.

$nin → Matches none of the values specified in an array.

$or → Joins query clauses with logical OR.

$and → —— || —— AND.

$not → inverts the effect of a query expression.

$nor → joins quey clause with a logical NOR.

Query 5 :-

- Count the no. of manufactures whose names have the partical string "am" in it.

```
db. Beers . find (
        name : { $ regrex/am/i }) . count ()
```

Query 6 :- same, but name starts with 'Am'.

- db. Beers . find (name : { $ regres /^Am/ } . count ()

Query 7 :- starts with "Am" ends with "corp"

- db. Beers . cout (name : { $ regrex : /^Am.* corp$$
  / }]

Array Operations :-

Q. Find items which are tagged as "popular" or "organic".

→
```
db.inventory.find(
    { tags : { $in : [ "popular", "organic" ] } )
```

Q. Find items which are not tagged as "popular" nor "organic".

→
```
db.inventory.find(
    { tags : { $nin : [ "popular", "organic" ] } )
```

Q. Find the 2nd and 3rd elements of tags.

-
  ```
  db.inventory.find( { },
        { tags : { $ slice : [1, 2] } } )
  ```
  ↗ skip count ↖ Return how many

-
  ```
  db.inventory.find( { },
        tags : { $ slice : -2 } )
  ```

Q. Find a document whose 2nd element in tags is "summer".

→
```
db.inventory.find ( tags.1 : "summer" )
```

Aggregation _ Function and Querying - Aerospike

Retrieving Big Data.
On counting and Distinct :-

→ Count the no. of Drinkers

- select count (*)
  FROM Drinkers.
- db. Drinkers. count ()

→ Count the no. of unique add^r of Drinkers.
- Select count(distint add^r)
  From Drinkers.
- db. Drinkers. count (add : { $ existrs : true })

→ Get the distint values of array
- Data : { _id : 1, places : [USA, France, USA, Spain, UK, S

- db. countryDB. distinct (places)
  - [USA, France, Spain, UK]

- db. country DB. distinct (places). length
  - 4

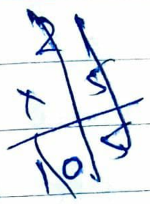
Aggregation Framework :-
→ Role of aggregation framework
- Grouping , aggregate functions , sorting ,

Multi-attribute Grouping

```
→ db. computers. aggregate (
  [
   {
    $group :{
 _id :{brand :"$brand",
       title : "$ title",
       category : "$ category",
        code : "$ code"},
 count : {$sum :1}
   }
 }
 {
    $ sort : { count : 1, category: -1}
 }
   ] )
```

* Text Search with Aggregation:-

```
      db. articles. aggregate (
  [
   {$match : {$text : {$search : "Hillary Democrat"}
   {$ sort : { score : {$ meta :" textscore"}}},
   {$ project : {title :1, _id :0}}
   ])
```



Notes

SELECT	projection
FROM	collection
Where	query filter

Retrieving Big Data Quiz

1. What does it mean for a query language to be declarative?
→ The language specifies what data to obtain.

2. Use the following table named "user ~~table~~" to answer the next 2 problems.

userId	username	email
1	admin	admin@corporate.moe
2	h4x0r	1337@rawr.cte

Q. How would u go about querying the entire username column?
>> SELECT username FROM user_table

3. How would u go about querying the entire database table
→ SELECT * FROM user_table

. global indexing table?
→ A index table in order to keep track of a given da
ype that might exists within multiple machines.

5. What are the three computing steps of semijoin?
→ project, ship, Reduce

6. What is purpose of a semijoin ?
→ Increase the efficiency of sending data across multiple machines.

7. What is subquery ?
→ A query statement within another query

8. What is a correlated subquery ?
→ A type of query that contain a subquery that requires information from a query one level up.

9. purpose of Group by queries
→ Enables ~~of~~ calculations based on specific colu
of the table.

13. Statement that we would need to grab email into for user indexes greater than 24 ?
→ db. email . find ({ userIndex : { $gt : 24 } },
{ email : 1 , -id: 0 })

Notes

14. What does it mean to have a _id : 0 within our query statement ?
→ Tell MongoDB not to return a document i