LAB - 1

PAGE NO :
DATE : 4/03/25

1. Create Database

use myDB;

db; // to list all databases

show dbs;

II CRUD

db.createCollection("Student");

db.Student.drop();

db.Student.insert({_id:1, StudName:"Michelle Jacintha", Grade:"VII", Hobbies:"Dance"});

db.Student.update({_id:3, StudName:"Aryan", Grade:"VII"}, {$set:{Hobbies:"SK"}}, {upsert:true})

db.Student.find({StudName:"Aryan"});

db.Student.find({}, {StudName:1, Grade:1, _id:0});

db.Student.find({Grade:{$eq:'VII'}}).pretty();

db.Student.find({StudName:/^N/}).pretty();

db.Student.find({StudName:/a/}).pretty();

db.Student.count();

PAGE NO :
DATE :

db.Student.find().sort({StudName:-1}).pretty()

Import data from a CSV file.

mongoimport --db mydb --collection Student
--type csv --headerline --file "C:/Admin/
downloads/demo.csv";

Export data to a CSV file

mongoexport --host localhost --db mydb
--collection Student --csv --out "C:/Admin/
document "Year", "Quarter"

PAGE NO :
DATE :

LAB - 2

1. use yourDatabaseName ;

// create and insert records.

```
db.Customers.insertMany([
    { Cust_id : 1, Acc_Bal : 1500, Acc_Type : 'z'},
    { Cust_id : 2, Acc_Bal : 500, Acc_Type : 'x'},
    { Cust_id : 3, Acc_Bal : 2000, Acc_Type : 'z'},
    { cust_id : 4, Acc_Bal : 3000, Acc_Type : 'y'},
    { cust_id : 5, Acc_Bal : 1200, Acc_Type : 'z'}
]);
```

// Query to display records where acc bal is greater than 1200 & acc type is 'z'.

```
db.Customers.find({ Acc_Bal : { $gt : 1200},
                    Acc_Type : 'z' });
```

// ~~Query~~ to determine the min & max acc bal for cust_id

```
db.Customers.aggregate([
    {
        $group : {
            _id : "$cust_id",
            min_balance : { $min : "$Acc_Bal"},
            max_balance : { $max : "$Acc_Bal"}
        }
    }
]);
```

PAGE NO :
DATE :

2

```
db.createCollection("Products");

db.Products.insertMany([
{ p_id: "123", name: "Sphone", category: "Ele",
  price: 299.9, quantity: 50},
{ p_id: "67890", name: "Laptop", Category:
"Ele", price: 899.9, quantity: 30},
  :
]);

db.createCollection("Users");

db.Users.insertMany([
  { user_id: "789ghi",
    name: "John",
    email: "abc@email.com",
    cart: [
        { p_id: "12345", quantity: 2},
        { p_id: "23456", quantity: 1}
      ],
    orders: [
     {
       order_id: "order123",
       o_date: ISO("2023-03-01T00:00:00
       products: [
         { p_id: "12345", quantity: 1,
           price: 299.99},
         { p_id: "23456", quantity: 2,
           price: 49.9}
       ],
       total_price: 399.9
     }], ...});
```

PAGE NO :
DATE :

MongodB Queries

```
db.Products.find({});  // Retrieve

db.Products.find({ category : "Ele" });
                    // Retrieve from "Ele"

db.Products.find({ quantity : { $gt : 0 } });
            // Retrieve products quantity > 0.

db.Products.find({}).sort({ price : 1 });
                    // Retrieve in assending order.

db.Products.find({ price : { $lte : 100 } });
                    // products with price <100.
```

Aggregation Queries:

```
db.Users.aggregate([
    { $match : { user_id : "789 ghi" } },
    { $unwind : "$cart" },
    { $lookup : {
        from : "Products",
        localField : "cart.p_id",
        foreignField : "p_id",
        as : "cart_items"
      }
    },
  { $project : { "cart_items.name" : 1,
                 "cart_items.price" : 1,
                 "cart.quantity" : 1 } }
]);
```

PAGE NO :
DATE :

1 db.Users.find({user_id:"123abc"},{orders:1})
// retrieve orders placed by user with id 123abc

2 db.Users.aggregate([
{$match:{user_id:"123abc"}},
{$unwind:"$orders"},
{$group:{_id:"$user_id",
total_spent:{$sum:"$order.total_price"}}}]);

Additional aggregation queries
// calculate the total price.

2 db.Products.aggregate([
{$group:{_id:"$category",total_product:{$sum:1}}}]);

2 db.Products.aggregate([
{$group:{_id:null,average_price:{$avg:"$price"}}}
]); // calculate average price.

3 db.Products.find({quantity:{$lt:10}});
// to find products with quantity <10

4 db.Products.find({}).sort({price:-1});
// Sort products in descending order.

5 db.Orders.aggregate([
{$group:{_id:"$user_id",total_spent:{$sum:"$total_price"}}}]);

PAGE NO :
DATE :

6 // To find users with highest total price

```
db.Orders.aggregate([
  { $group: { _id: "$user_id", total_spent:
    { $sum: "$total_price" } } },
  { $sort: { total_spent: -1 } },
  { $limit: 1 }
]);
```

7 // To find the average total price of orders

```
db.Orders.aggregate([
  {
    $group: { _id: null, average_order_price:
      { $avg: "$total_price" } }
  }
]);
```

8 // Calculate Total No. of Products in each category

```
db.Products.aggregate([
  { $group: { _id: "$category", total_prod:
    { $sum: 1 } } }
]);
```

S. Reddy

PAGE NO :
DATE :

1/04/25

Create keyspace:

> Create keyspace students with replication = {'class': 'SimpleStrategy', 'replication_factor': 1};

Describe keyspaces:

select * from system_schema.keyspaces;

Use students;

create table students_info (Roll_No int primary key, studname text, dob timestamp, last_exam_perc double);

describe tables;

describe tables students_info;

> CRUD

Begin Batch
Insert into students_info (Roll_no, studname, dob, last_exam_perc)
values (1, 'asha', '2012-03-12', 79.9)
Insert into students_info (Roll_no, studname, dob, last_exam_perc)
values (2, 'Kiran', '2012-03-12', 89.9)
Insert into students_info (Roll_no, studname, dob, last_exam_perc)
values (1, 'Tarun', '2012-03-12', 78.9)

PAGE NO :
DATE :

Apply Batch;

select * From students_info;

Roll_no	dob	last_exam_perc	studname
1	2012-03-12	79.9	asha
2	2012-03-12	89.9	kiran
3	2012-03-12	78.9	Tarun

select * from students_info where Roll_no in(1,2);

select * from students_info where studname='asha';

create index on students_info (studname);

sled Roll_no, studname from students_info limit 2;

Select Roll_no as "USN" from students_info;

→

USN
1
2
3

update students_info set studname='David' where Roll_no=2;

delete last_exam_perc from students_info where Roll_no=2;

Alter table students_info add hobby set<text>

PAGE NO :
DATE :

update students_info
set hobby = hobby + {"chess, "tennis'}
where Roll_no = 1;

create table library (counter_value =
counter_value + 1 where book_name = 'BDA'
and studname = 'jeet';

create table user login (id int primary key,
pss text);

Insert into userlogin (id, pss) values
(1, 'infy') using TTL 30;

Select TTL (pss) from userlogin where
user id = 1;

output :-

TTL (pss)
28

Roll_no	dob	hobby	language	last_exam_per
1	2012-03-11	{'chess TT'}	null	99.9

studname
asha

1/A/25

PAGE NO :
DATE : 8/04/25

LAB-03 Cassandra Usage

$ cqlsh

cqlsh > use Library;

```
cqlsh : library > create table library_info
        (stud_Id int, stud_Name text,
        Book_name text, Date_of_Issue
        date, Primary key (Stud_Id, Book_Id
        ));
```

```
cqlsh : library > create table Book_Counter
        (Stud_Id int, Book_Name text,
        Counter_value counter, primary
        Key (Stud_Id, Book_name));
```

```
cqlsh : library > begin batch
... insert into library_Info (Stud_Id,
    stud_Name, Book_name, Book_Id, D_O_I)
    values (112, 'alice', 'bda', 'b101', '08-07-
    2025');
... apply batch;
```

```
cqlsh : library > select * from library_Info;
```

~~cqlsh : library > se~~

stud_Id	book_id	book_name	D-O-I	stud
113	b102	ads	2025-04-07	bob
112	b101	bda	2025-04-07	alice

```
cqlsh : library > update Book_Counter set
        Counter_value = Counter_value + 1 where
        Stud_Id = 112 and Book_name = 'bda';
```

stud_id	book_id	book_name	date_of_issue	stud_name
113	b102	ads	2025-04-07	bob
112	b101	ada	2025-04-07	alice

cqlsh: library> select * from Book_Counter;

stud_id	book_name	counter_value
113	ads	1
112	bda	3

cqlsh: library> cqlsh -e "Copy library.Library_Info To 'library_info.csv' with Header = TRUE;"

cqlsh: library> select * from Library_info;

stud_id	book_id	book_name	date_of_issue	st_name
114	B103	ML	2024-04-03	John
117	B106	AI	204-04-04	Meena
120	B104	Cloud	2024-04-06	Arjun
123	B105	Bigdata	2024-04-07	Vikram
118	B103	ML	2024-04-05	Amit

S.P. 8/4/25

LAB-06

PAGE NO :
DATE : 15/04/25

Hadoop Usage

> $ start-all.sh

mkdir> $ hdfs dfs -mkdir /demo208

ls > $ hadoop fs -ls /
Found 1 items
drwxr-xr-x - hadoop supergroup 0 2023-04-15 15:00 /demo208

put > $ hdfs dfs -put /home/hadoop/Documents/demo.txt /demo208/file.txt

cat > $ hdfs dfs -cat /demo208/file.txt
This is the demo for the usage of put method in hadop

copyFromLocal
> $ hdfs dfs -copyFromLocal /home/hadoop/Documents/demo.txt /demo208/file_cp.txt

cat > $ hdfs dfs -cat /demo208/file_cp.txt
This is the demo for the usage of put method in hadoop.

get > $ hdfs dfs -get /demo208/file.txt /home/hadoop/Documents/demo.txt

=> File exists.

PAGE NO :
DATE :

getmerge > $ hdfs dfs -getmerge /demo208/file.txt /demo208/file_cp.txt /home/hadoop/Documents/demo.txt

getfacl > $ hadoop fs -getfacl /demo2081
file : /demo208
owner : hadoop
group : supergroup
user :: rwx
group :: r-x
other :: r-x

copyTolocal $ hdfs dfs -copyTolocal /demo208/file.txt /home/hadoop/Documents

move > $ hadoop fs -mv /demo208 /abc

ls > $ hadoop fs -ls /abs

Found 2 items
-rw-r--r-- 1 hadoop supergroup 55 /ab/file.txt
-rw-r--r-- 1 hadoop supergroup 55 /abc/file.txt

S. Priya
17/4/25

PAGE NO :
DATE :

LAB - 05 WordCount Hadoop

WCDriver:

```
package wordcount;

import java.io.IOException;

public class WCDriver extends Configured
    implements Tool
{
    public int run(String args[]) throws
        IOException
    {
        if (args.length < 2)
        {
            System.out.println("give inputs");
            return -1;
        }
        JobConf conf = new JobConf(WCDriver.class);
        FileInputFormat.setInputPaths
        FileOutputFormat.setOutpath(conf, new Path
                                    (args[1]));
        conf.setMapperClass(WCMapper.class);
        conf.setReducerClass(WCReducer.class);
        conf.setOutputValueClass(Text.class);
        conf.setOutputKeyClass(IntWritable.class);
        JobClient.runJob(conf);

        return 0;
    }
}
```

PAGE NO :
DATE :

WCMapper.java

```
import java.io.IOException;

public class WCMapper extends MapReduceBase
implements Mapper <LongWritable, Text, Text,
output, Reporter rep) throws IOException
{
    String line = value.toString();
    for (String word : line.split(" "))
    {
        if (word.length()>0)
        {
            output.collect(new Text(word))
        }
    }
}
```

WCReducer.java.

```
package wordcount.

import java.io.Exception;

public class WCReducer extends MapReduceBase
implements Reducer <Text, IntWritable>

public void reduce (Text key, Iterator
<IntWritable> value, OutputCollector<Text
IntWritable>
    int count = 0
```

```
while (value.hasNext())
{
    IntWritable i = value.next();
    count += i.get();
}
output.collect(key, new IntWritable(count));
}
}
```

output:

```
are 1
brother 1
cat 1
family 1
hi 1
how 1
is 4
job 1
like 1
you 1
your 4
```

PAGE NO :
DATE :

LAB 06 FindAverage temperature

AVDriver.java

```
package WAverage;

import org.apache.hadoop.fs.Path;

public class AVDriver
{
    public static void main (String[] args
    {
        if (args.length != 2)
        {
            System.err.println ("Enter");
            System.exit(-1);
        }

        job.setJarByClass (AVDriver.class);
        job.setJobName ("Max temperature");

        FileInputFormat.addInputPath(
                job.new Path (args[0]));
        FileOutputFormat.setOutpath
                (job, new Path (args[1]));

        job.setOutputKeyClass (Text.class);
        job.setOutputValueClass(IntWrit
                .class);
    }
}
```

PAGE NO :
DATE :

AVMapper.java

```
package MAverage.

import java.io.IOException;

public class AVMapper extends Mapper<
  LongWritter, Text, IntWritable>
{

public static final int Missing = 9999;

public void map (LongWritable key,
  Text value)

int temp;

String line = value.toString();
string year = line.substring(15,19);

if (line.charAt(87) == '+')
{
  temp = Int.parseInt(line.substring
                (88,92));
string quality = line.substring(92,93);

if (temp != 9999 && quality.matches
                ("[01459]"))
  context.write (new Text(year), new
  IntWritable (temperature));
}
}
```

PAGE NO :
DATE :

AVReducer.java.

```
package WAverage;

import java.io.IOException;

public class AVReducer extends Reducer
    <Text, IntWritable, Text>
{
    public void reduce (Text key, iterable
        < IntWritable> values,

Text, IntWritable >. (context context)
throws InterruptedException
{
    int max=0,
    int count=0;
    for (IntWritable value : values)
    { max_temp += value.get();
        count++;
    }
    context.write (key, new IntWritable
            (max_temp/count));
}}
```

output:

1901 46

LAB - 06. Find Mean Max Temperature

MNMapper.java.

```
package Mean;

import java.io.IOException;

public class MNMapper extends Mapper
  <LongWritable, Text, Text, IntWritable>
  {
    public static final int Miss = 9999;

  public void map (LongWritable key,
    Text value, Mapper < longWritable,
    Text, Text >

    int temp;
    string line = value.toString();
    string month = line.substring(19,21);

    if (line.charAt(87) == '+')
    {
        temp = Int.parseI (line.substring
                                (88.92));
    string q = line.substring (92,93);

    if (temp ! = Miss)
      {
          write (new Text(month));
      }
  }
}
```

PAGE NO :
DATE :

MNReducer.java.

```
package Mean;

import java.io.IOException;

public class MNReducer extends Reducer.

public void reducer (Text key, Iterable
                    <IntWritable>
int max = Int.Min.
int total_temp = 0;
int count = 0;
int days = 0;

for (IntW val : vals)
{
    int t = val.get();
    if (t > max)
    {
        max = temp;
    }
    c++

    if (c == 3)
    {
        t. + = max;
        max = Int.Min
        c = 0
        d++;
    }
}
```

LAB - 07 Find list of TopN

PAGE NO :
DATE :

TNMapper.java.

```
Package TopN;

import java.io.IOException;

public class TNMapper extends Mapper
{
    private final static IntW one = new
                                    IntW(1);
    private Text word = new Text();

    public void map()
        String cleanL = val.toString();
        String Tokenize = ne StringTokan();

    while( itr.hasMoToken())
    {
        String t = itr.nextToken().trim();
        if( ! token.isEmpty())
        {
            word.set(token);
            context.write(word, one);
        }
    }
}
```

PAGE NO :
DATE :

TNReducer.java

```
package ~~jaw~~.topN;

import java.io.IOException;

public class TNReducer extends Reducer

{  public void reduce()
     int s=0
     for(int W v = vs)
     {
        sum = v.get();
        countmap.put(key.toString());
     }

protected void cleanup(Context)
{
   int ~~to~~ = 20;
   int c = 0
   for(Entry<String, Integer>)
     {
        if(count==tN)
          break
     }
  }
}
```

output: banana 5
apple 4
fruit 3
mango 2
kiwi 1

LAB-08.

PAGE NO :
DATE :

Scala prgr to print 1 to 100

```
for ( i ← 1 to 100)
  { print ln (i) }
```

output:

1
2
3
4
5
6
7
8
.
.
.
100

PAGE NO :
DATE :

LAB - 9

Spark program to list words whose count is greater than 4. using RDD and flatmap.

```
echo "hello spark hello world spark
      spark code code code code spark"
      > input.txt.

> spark-shell

> val lines = sc.textFile("input.txt")
> val words = lines.flatMap(line => line.
                  split(" "))
> val wordPairs = words.map(word => (wor
                                  1))
> val wordCounts = wordPairs.reduceByKey
                                  (_+_)
> val frequentWords = wordCounts.filter
          {case(wor, count) => count>4}

> frequentWords.collect().foreach(println)
```

output :

```
code 5
spark 5
```

PAGE NO :
DATE :

LAB - 10.

Write a simple streaming prgrm in spark text data streams on a particular port perform basic text cleaning & print the cleared text.

```
def commotize (word : string) : string :word.
(match (c) <
        case w if w.endswith (ing) =>
         w.stuffle (c)
        case => word
>

val clear text = def (line : string)
      => <
        line to. lowercase
      .re replace. all ("t"a - z "\\ 5]", ",")
      . split ("5t")
        . map (lematize)
        - make strings (" ")
    >)

    val lin = spark. readstream.
    val very = ("socket")
      option ("host", "local/host")
       - load()
    val cleared. ("value")
      query. await. remintion ).
```