LOK JAGRUTI KENDRA UNIVERSITY

Day wise schedule of MCA, 2nd Semester

Course Name: Big Data Tools (BDT)

Unit-1	Day 1
	Detailed Schedule
1.	Introduction to Big Data and analytics classification of digital data, Structured and Unstructured Data. Why Big Data? Traditional business intelligence versus Big Data.
Essential Assignment	
1.	Overview of current big data tools & technologies **MongoDB:** MongoDB overview, MongoDB installation **MongoDB shell commands / queries:** view all databases, create new database, drop existing database, view current database, and switch over to a given database, db. help(), display statistics of a given database, display current version of MongoDB server, display list of collections in current database. Create collection, drop collection, data types, insert document, update document, delete document
2.	Create a new MongoDB database called "TEACHER". Within "TEACHER" database, create a collection named "TEACHER_MASTER". Assume an appropriate Schema consisting of fields like Name, Age, Subject(array), DOB, Gender, Salary, City 1. Insert 7 documents into the above collection. 2. Display Name, Subject, Gender and Salary. 3. Display teacher, which are from the city "Ahmedabad". 4. Display the teacher id, name, city and DOB. 5. Display the teachers whose gender is female and teach either "Hindi" or "English" subject. 6. Update all those documents where name of teacher is "Anil" with the new value of subject as "English". 7. Delete data of all those teachers who were born after 1st January 1980. 8. Remove field age. 9. Display the teachers that do not teach "English" subject and their salary is more than 30000. 10. Find all the teachers having gender "Male" and display salary for them.
Desirable Assignment	

1.	Create a new MongoDB database called “EMPLOYEE”. Within “EMPLOYEE” database, create a collection named EMPLOYEE_MASTER” assume an appropriate schema consisting of fields like Empno, Name, Designation, DOJ, Department, Salary, Gender, Skills(array) 1. Insert 7 documents into the above collection. 2. Display Name, Department, Gender and Salary. 3. Display the list of employees from the department “HR”. 4. Display the employee id, name, department and DOB. 5. Display the employees whose gender is female and designation either “Engineer” or “Scientist”. 6. Update all those documents where name of employee is “Akash” with the new value of designation as “Engineer”. 7. Delete all those documents where DOJ is before 1st January 1999. 8. Display the employees whose salary is greater than 25000 and have skills Java or PHP. 9. Find all the employees having designation “Engineer” and display salary for them. 10. Display only those documents where the name of employee is “Amit” and designation is “Accountant”.

Unit-1	Day 2

Detailed Schedule

1.	CRUD operations in MongoDB: Creating Document Oriented Data – Schema and collections Insert document, query document, projection, limiting record, sorting record

Essential Assignment

1.

Create a Student Master database with a collection called "Student" containing documents with some or all of the following fields: StudentRollNo, StudentName, Grade, Hobbies, and DOJ.
Perform the following operations on the database:

a) Insert 10 Records in the database.
b) Find the document where in the "StudName" has value "Akhay".
c) Find all documents in proper (like tabular) format. (Without _Id field)
d) Retrieve only Student Name and Grade.
e) Retrieve Student Name and Grade of student who is having _id column is 1.
f) Add new field "Address" in Student collection.
g) Find those documents where the Grade is set to 'VII'.
h) Find those documents where the Grade is not set to 'VII'.
i) Find those documents where the Hobbies is set to either 'Chess' or is set to 'Dancing".
j) Find those documents where the Hobbies is set neither to 'Chess' nor is set to 'Dancing".

2.

Create a Movie_Maker database with a collection called "Movie" containing

documents with some or all of the following fields: titles, directors, years, actors.

Perform the following operations on the database:

a) Retrieve all documents.

	b) Retrieve all documents with Director set to "Quentin Tarantino". c) Retrieve all documents where actors include "Brad Pitt". d) Retrieve all movies released before the year 2000 or after 2010. e) Add a synopsis to "The Hobbit: An Unexpected Journey": "A reluctant hobbit, Bilbo Baggins, sets out to the Lonely Mountain with a spirited group of dwarves to reclaim their mountain home - and the gold within it - from the dragon Smaug." f) Add a synopsis to "The Hobbit: The Desolation of Smaug": "The dwarves, along with Bilbo Baggins and Gandalf the Grey, continue their quest to reclaim Erebor, their homeland, from Smaug. Bilbo Baggins is in possession of a mysterious and magical ring." g) Add an actor named "Samuel L. Jackson" to the movie "Pulp Fiction" h) Find all movies that have a synopsis that contains the word "Bilbo". i) Find all movies that have a synopsis that contains the word "Gandalf". j) Find all movies that have a synopsis that contains the word "Bilbo" and not the word "Gandalf". k) Find all movies that have a synopsis that contains the word "dwarves" or "hobbit" l) Find all movies that have a synopsis that contains the word "gold" and "dragon". m) Delete the movie "Pee Wee Herman's Big Adventure"
	Desirable Assignment
1.	The following collection of five documents is given. Documents consist of orders. An order has an id (e.g. "o1"), the year in which it was issues, the cost, the items in the order, and the number of days it took to deliver the order. The cost is specified as price in a given currency. The order items consist of products. A product has an id (e.g., "p1"), colors, and quantity. {"order":"o1","year":2020,"paid":"Y","cost":{"price":30,"currency":"NOK"}, "items":[{"product":"p1","colours":["blue","black"],"quantity":15}],"delivery_days":5}

{"order":"o2","year":2020,"paid":"Y","cost":{"price":13,"currency":"EUR"},
"items":[{"product":"p2","colours":["white"],"quantity":4},
{"product":"p3","colours":["white","black"],"quantity":1}],"delivery_days":4},
{"order":"o3","year":2018,"paid":"N","cost":{"price":33,"currency":"EUR"},
"items":[{"product":"p3","colours":["blue","black"],"quantity":4}], "delivery_days":4},
{"order":"o4","year":2017,"paid":"Y","cost":{"price":17,"currency":"NOK"},
"items":[{"product":"p2","colours":["pink","black"],"quantity":14},
{"product":"p4","colours":["white"],"quantity":1}], "delivery_days":2},
{"order":"o5","year":2020,"paid":"Y","cost":{"price":19,"currency":"NOK"},
"items":[{"product":"p1","quantity":15}], "delivery_days":3}

1. Retrieve all documents in a collection
2. Retrieve all documents that contain paid orders (the "paid" field is "Y")
3. Retrieve all documents that contain paid orders and the orders are from before 2019
4. Retrieve all documents that contain unpaid orders or whose orders are from before 2019
5. Retrieve all documents that contain orders whose price is in NOK
6. Retrieve all documents that contain orders whose price is less than 18 NOK
7. Retrieve all documents with orders that contain product "p2"
8. Retrieve all documents with orders that contain products whose quantity is less than 13.
9. Retrieve all documents with orders that contain products whose quantity is less than 13 and contain no products whose quantity exceeds 13
10. Retrieve all documents with orders that contain products whose first colour (i.e., first element in the "colours" array) is blue
11. Retrieve the total number of delivery days, grouped by year; retrieve the results only after 2017 (Hint: use aggregation pipelines)

Unit-1	Day 3
.	**Detailed Schedule**
1.	MongoDB query document(using find()) method with examples, Add MongoDB array using insert(), MongoDB primary key, MongoDB count() & remove() functions, MongoDB regular expression, MongoDB Vs. SQL
Essential Assignment	
1.	**Refer Student Master database example of Day-2.** k) Find those documents where the student name begins with 'M'. l) Find those documents where the student name has an "e" in any position. m) Find those documents where the student name ends in "a". n) Find total number of documents. o) Find total number of documents where Grade is 'VII'. p) Sort the documents in ascending order of student name. q) Display the last two records.
2	Create database EMP and Make Collection With name "EMPL" and Follow Queries Insert Records Into EMPL Collection {no:1,name:"ST",salary:2000,role:"OB"}, {no:2,name:"MSD",salary:1500,role:"WK"}, {no:3,name:"YS",salary:1000,role:"ALR"}, {no:4,name:"RD",salary:1000,role:"MOB"}, {no:5,name:"RS",salary:500,role:"OB"}, {no:6,name:"BK",salary:500,role:"MOB"}, {no:7,name:"VK",salary:300,role:"BW"}, {no:8,name:"JB",salary:400,role:"BW"}, {no:9,name:"HP",salary:400,role:"ALR"}, {no:10,name:"VS",salary:300,role:"OB"}]) 1. Display data in tabular format. 2. Update salary of employee where name is "ST" also increase salary by +8000. 3. Update salary of all employee by giving an increment of +4000 each employee. 4. Update role of employee whose name is "MSD" as "C and WK" 5. Add a new field remark to document with name "RS" set remark as WC 6. Add a new field as number "11", name "AK", Salary "10000", role "coach" without using insert statement, but for doing so you should have a record added

	number 11. 7. Remove added new field(number “11”). 8. Update the field "RD" by multiplying with salary by 2. 9. Find document from the empl collection where name begins with ‘S’ 10. Find document from the empl collection where name begins with ‘R’ 11. Find document from the empl collection where name ends with ‘K’ 12. Find document from the empl collection where name ends with ‘D’ 13. Find document from the empl collection where name has S in any position ———————————————————————— Regular Expression ———————————————————————— (Note: Use Case sensitive allow For that write in Option: "i") 1. Find document from the empl collection where name begins with ‘S’ 2. Find document from the empl collection where name begins with ‘S’ ———————————————————————— Use of $in and $nin (in and notin) ———————————————————————— (Note: There will not use {} braces in that $in and $nin) 1. Display documents where in empl collection field have OB,MOB 2. Display documents where in empl collection field not have OB,MOB
Desirable assignment	
1.	Create a “Showroom” database with a collection called “Cars”, containing documents with some or all of the following fields: Model No, Car Name, Colors (Note: An array is expected), Speed, Mfd Country, and Date of manufacturing (Note: Use Proper Date format). Perform the following operations on the database. (Insert at least 10 documents) 1. Display only the car names, where speed is greater than 50. 2. Find the documents, where Model No is either “Md0135” or “Md0453”. 3. Display the names and date of manufacturing of cars, whose speed is greater than 40 and less than 65. 4. Sort all the documents based on the Car Name in the descending order. Display only the first 5 records.

	5. Display the last 3 records where Mfd Country is “India”. 6. Add a field called “Descriptor”, which describes the car model for documents with Model No, is “Md0122”, “Md0255” and “Md0135”. 7. Search the documents where the Descriptor field contains the word “popular” or “famous”. 8. Find the Model No’s and Car Names of those documents where the colours have “red” and “green” as elements.

Unit-1	Day 4
	Detailed Schedule
1.	**MongoDB Revision:** Dealing with Using NULL Values, Count, Limit, Sort, Skip, Arrays and Array Operations, Aggregate
Essential Assignment	
1.	Create a database named "Store" in MongoDB with a collection called "Sales" containing documents with some or all of the following fields: customerId, customerName, gender, DOB, contactNumber, address (containing fields: houseNo, street, area, city, pincode), orders (containing fields: orderId, orderDate, items (containing fields: itemId, itemName, itemPrice, quantityOrdered, discount)). Note that some customers may not provide their date of birth and/or contact number. Also, not all products would be sold at a discount. Perform the following operations on the database (either in the console or using any programming language): a) Insert records for 3 customers and 5 items in at least 20 orders. b) Update the contact number of a custome"1234567890" whose customerName is "Smith" . c) Display customerId, customerName, gender, contactNumber, of customers residing in "Ahmedabad". d) Display city-wise count of customers
2	Create a database named "Inventory" in MongoDB with Collection "Sales" to accommodate the following fields: productId, productName, saleDate (Note: Use Proper Date format), salePrice, saleQuantity, purchaseDate (Note: Use Proper Date format), purchasePrice, purchaseQuantity. Insert at least 10 records. Perform the following operations on the database: 1. Display only the productName, where salePrice is less than 150. 2. Change the product name of TV to Television. 3. Add a field productsize (an array) with values "small", "middle" and "big" for document with productId "Pid0023" and "Pid0231". 4. Group on productId and compute the maximum of salePrice. 5. Display the productName and saleDate, whose saleQuantity is greater than 100 and less than 165. 6. Add a field called "Descriptor", which describes the product for documents

	with Model No, is “Pid0044”, “Pid0231” and “Pid3211”. 7. Search the documents where the Descriptor field contains the word “healthy” and “fruit”. 8. Create Index on productID. 9. Remove field salesDate where productID is “Pid0555”.
3	Create a database named “BookStore” in MongoDB with a collection called “Books” containing documents with some or all of the following fields: bookId, bookTitle, authors (containing fields: authorName), publicationYear, publisher, Orders (containing fields: OrderedId, orderDate, customerName, price, quantityOrdered, discount). Note that a book may have one or more authors and orders. Also, the same Ordered can be present in one or more books. Perform the following operations on the database (either in the console or using any programming language): a) Insert records for 10 books from 5 authors, and at least 20 orders in total. b) Update the title of a particular book. c) Display all the books having less than 3 authors and sort by book name. d) Display the number of books from each publisher
Desirable Assignment	
1	Create “Mymenu” database with a collection called “Restaurants”, containing documents with some or all of the following fields: Restaurant Id, Restaurant Name, Grades (Note: An array is expected), Cuisine, Address (Note: Must include Building Name, Street, Area, City, ZipCode), and Date of Establishment (Note: Use Proper Date format), Score and Rating. Perform the following operations on the database. (Insert at least 10 documents) 1. Find the Restaurant Names, who have established after January 2010. 2. Find the restaurants that do not prepare Cuisine of “American”, and their Score is more than 70. 3. Update the Rating of the restaurant “Mirch Masala”. 4. Display the restaurants, which are located in “Ahmedabad”. 5. Find the Restaurant Names and Cuisine, for those restaurants which contain ‘chen’ (Example: “Kitchen”) as the last three letters. 6. Find the Restaurant Id’s and Restaurant Names of those restaurants, which are situated in “Ahmedabad” (City) but not in “ISKON” (Area). 7. Add a field “Borough” with value “Bronx”, for restaurants with _id: 3 and 4. 8. Remove the field Cuisine for restaurants whose name is “Jassi De Paratha”. Remove the document, with restaurant named “Barbeque Nation”.

2	Write a MongoDB query to display the fields restaurant_id, name, borough and zip code, but exclude the field _id for all the documents in the collection restaurant.
	Structure of 'restaurants' collection :
	{
	"address": {
	"building": "1007",
	"coord": [-73.856077, 40.848447],
	"street": "Morris Park Ave",
	"zipcode": "10462"
	},
	"borough": "Bronx",
	"cuisine": "Bakery",
	"grades": [
	{ "date": { "$date": 1393804800000 }, "grade": "A", "score": 2 },
	{ "date": { "$date": 1378857600000 }, "grade": "A", "score": 6 },
	{ "date": { "$date": 1358985600000 }, "grade": "A", "score": 10 },
	{ "date": { "$date": 1322006400000 }, "grade": "A", "score": 9 },
	{ "date": { "$date": 1299715200000 }, "grade": "B", "score": 14 }
	],
	"name": "Morris Park Bake Shop",
	"restaurantId": "30075445"
	}

3	Book name, isbn number, author name(can be more than one author), issue date, return date, price, member identification number, book category (it can be reference book, text book, journal, magazine), status (1 for available and 0 for not available) Perform the following queries using MongoDB database 1. List book details which is issued to member “MC01” 2. How many books for “computer network” and “ network security” 3. Update the library document and remove return date field. 4. List total number of books category wise 5. List all books, which is either textbook or reference book. 6. Sort the document based on price (book having max price comes first)

	7. Count total number of books which is not available for ‘journal’ category 8. Create an index for library collection.

Unit-2	Day 5
.	**Detailed Schedule**
1.	**MapReduce:** What is Map Reduce Programming? How does Map Reduce Works? Map Reduce Word Count Example . About Map Reduce ,Understanding block and input splits, MapReduce Data types ,Understanding Writable ,Data Flow in MapReduce Application , Understanding MapReduce problem on datasets , MapReduce and Functional Programming , Writing MapReduce Application , Understanding Mapper function , Understanding Reducer Function , Understanding Driver , Usage of Combiner
Essential Assignment	
1.	Create “Mymenu” database with a collection called “Restaurants”, containing documents with some or all of the following fields: Restaurant Id, Restaurant Name, Grades (Note: An array is expected), Cuisine, Address (Note: Must include Building Name, Street, Area, City, ZipCode), and Date of Establishment (Note: Use Proper Date format), Score and Rating. Perform the following operations on the database. (Insert at least 10 documents) Use MapReduce function to display the total Rating for every restaurant (Note: Group on Restaurant Id), whose Score is less than 50.
2	Find Out total number of posts by each user whose status is active. use Blogs db.createCollection("Blogs") db.Blogs.insert([{_id:1,userId:"u101",post_text:"Python MapReduce", status:"Active"}, {_id:2,userId:"u101",post_text:"Python Socket", status:"Active"}, {_id:3,userId:"u102",post_text:"Java Script", status:"Active"}, {_id:4,userId:"u102",post_text:"Css&bootstrap", status:"Active"}, {_id:5,userId:"u101",post_text:"Python ADT", status:"Passive"},])
3.	db.Customers.insert({CustID:"C123", AccBal:500, AccType:"S"}) db.Customers.insert({CustID:"C123", AccBal:900, AccType:"S"})

	db.Customers.insert({CustID:"C123", AccBal:1500, AccType:"C"}) db.Customers.insert({CustID:"C111", AccBal:1200, AccType:"S"}) db.Customers.insert({CustID:"C111", AccBal:1000, AccType:"C"}) db.Customers.insert({CustID:"C111", AccBal:5000, AccType:"S"}) Find Out total balance, Average balance of the customer whose AccType is "S".
Desirable Assignment	

1.	Orders- Collection:
	{
	_id:
	Cust_id:
	Ord_date:
	Price:
	items:[{sku: , qty: , price: },
	{ sku: , qty: , price: }]
	Status:
	}
	1. To find total Price per customer
	2. To find total orders and total quantity per customer
	(Note: sku-> stock keeping unit,qty->quantity)

2.	Create a new MONGODB database called "TEACHER". Within "TEACHER" database, create a collection named "TEACHER_MASTER". Assume an appropriate Schema consisting of fields like Name, Age, Subject(array), DOB, Gender, Salary, City. 1. Insert 7 documents into the above collection. 2. Perform a MapReduce function using MongoDB to Count Average Salary by Teacher.

Unit-2	Day 6
.	**Detailed Schedule**
1.	NoSQL: What is a NoSQL Database? Brief History of NoSQL Databases, NoSQL Database Features, Types of NoSQL Database (Document databases, Key-value databases, Wide-column stores, and Graph databases), Difference between RDBMS and NoSQL, Why NoSQL? When should NoSQL be used? NoSQL Database Misconceptions.
Essential Assignment	
1.	**MapReduce:** Create a database named "Store" in MongoDB with a collection called "Sales" containing documents with some or all of the following fields: customerId, customerName, gender, dataOfBirth, contactNumber, address (containing fields: houseNo, street, area, city, pincode), orders (containing fields: orderId, orderDate), items (containing fields: itemId, itemName, itemPrice, quantityOrdered, discount)). Note that some customers may not provide their date of birth and/or contact number. Also, not all products would be sold at a discount. Perform the following operations on the database (either in the console or using any programming language): a) Insert records for 3 customers and 5 items in at least 20 orders. b) Update the contact number of a particular customer. c) Display customerId, customerName, gender, contactNumber, of customers residing in "Ahmedabad". d) Display city-wise count of customers e) Use MapReduce function to display the number of times each item was sold.
2.	Create a database named "BookStore" in MongoDB with a collection called "Books" containing documents with some or all of the following fields: bookId, bookTitle,authors(containing fields: authorName), publicationYear, publisher, Orders(containing fields: OrderedId, orderDate, customerName, price,quantityOrdered, discount).

	1. Use MapReduce function to display the total quantity of books ordered for each date. 2. Use Map Reduce function to display the discount offered to a particular customer.
Desirable Assignment	
1.	Create a database “BookStore” with a collection called “Books” containing documents with some or all of the following fields: Category, BookName, Author, quantity, price, pages. Perform the following operations on the database: a) Insert Records for 5 books. b) Write Map & Reduce functions to split the books into the following two categories: Bigbooks, Smallbooks. (Books which have more than 300 pages should be in the Big books category. Books which have less than 300 pages should be in the Small books category.) c) Count the number of books in each category d) Store the output as follow as documents in a new collection called “Book

Book Category	**Count of the Books**
Big books	2
Small books	3

Unit-3	Day 7

Detailed Schedule

1.	Hadoop: Introducing Hadoop, File System - Concepts Blocks, Replication Factor, Version File ,Safe mode, Namespace IDs, Purpose of Name Node , Purpose of Data Node, Purpose of Secondary Name Node, Purpose of Job Tracker , Purpose of Task Tracker. **HDFS Shell Commands** – copy, delete, create directories etc. • **hdfs dfs -ls /** • **hdfs dfs -ls -d /hadoop** • **hdfs dfs -ls -h /data** • **hdfs dfs -ls -R /hadoop** • **hdfs dfs -ls /hadoop/dat*** • **hdfs dfs -text /hadoop/derby.log** • **hdfs dfs -cat /hadoop/test** • **hdfs dfs -appendToFile /home/ubuntu/test1 /hadoop/text2** • **hdfs dfs -put /sample /hadoop** • **hdfs dfs -put -f /sample /hadoop** • **hdfs dfs -put -l /sample /hadoop** • **hdfs dfs -put -p /sample /hadoop** • **hdfs dfs -get /newfile** • **hdfs dfs -get -p /newfile** • **hdfs dfs -get /hadoop/*.txt** • **hdfs dfs -copyFromLocal /sample /hadoop** • **hdfs dfs -copyToLocal /newfile** • **hdfs dfs -moveFromLocal /sample /hadoop**
Essential Assignment	
1.	HDFS Commands: -ls, -ls -R, -mkdir, -put, -get 1) Create a file “Sample” in a local file system and export it to the HDFS File System.

	2) Write the HDFS command for copying a “Sample” file from HDFS to local File System. 3) Write HDFS commands for creating “Test” directory in HDFS and then removing that directory. 4) Write HDFS command to display complete list of directories and files of HDFS. 5) Write HDFS command for displaying the contents of “Sample” text file in HDFS on screen. 6) Write HDFS command for copying an existing “Sample” file in a “Test” HDFS directory to some another HDFS directory.
Desirable Assignment	
1.	Practice HDFS command 1. Execute the HDFS command for getting the list of complete directories and files of HDFS. 2. Execute the HDFS command for displaying the contents of some Xyz. text file in HDFS on screen. 3. Execute the HDFS command for copying an existing sample file in a given HDFS directory to some another HDFS directory

Unit-3	Day 8
.	**Detailed Schedule**
1.	Reading and Writing in HDFS , Difference of Unix Commands and HDFS commands, Hadoop file command, Admin Commands overview , Hands on exercise with HDFS commands • hdfs dfs -cp /hadoop/file1 /hadoop1 • hdfs dfs -cp -p /hadoop/file1 /hadoop1 • hdfs dfs -cp -f /hadoop/file1 /hadoop1 • hdfs dfs -mv /hadoop/file1 /hadoop1 • hdfs dfs -rm /hadoop/file1 • hdfs dfs -rm -r /hadoop hdfs dfs -rm -R /hadoop hdfs dfs -rmr /hadoop • hdfs dfs -rm -skipTrash /hadoop • hdfs dfs -rm -f /hadoop • hdfs dfs -rmdir /hadoop1 • hdfs dfs -mkdir /hadoop2 • hdfs dfs -mkdir -f /hadoop2 • hdfs dfs -touchz /hadoop3
Essential Assignment	
1.	HDFS Commands: -copyFromLocal, -copyToLocal, -cat, -cp, -rm-r To get the list of all the files in the HDFS root directory 1. Help 2. Write a command to Listing all the files in HDFS. 3. Write a command to copy of a file “Abc.txt” from Local file System to Hadoop FS. **Inserting Data into HDFS** Step1: Create an input directory Step2: Use the put command transfer and store the data file from the local systems to the HDFS using the following commands in the terminal.

	Step3: Verify the file using ls command. **Retrieving Data from HDFS** **Step1:** View the data from HDFS using the cat command. **Step2**: Gets the file from HDFS to the local file system using get command as shown below
Desirable Assignment	
1.	Practice HDFS command 1. Taking any data/file of your choice, execute the HDFS command for copying a given sample file in local file system to HDFS. 2. Taking any data/file of your choice, execute the HDFS command for copying a given sample file in HDFS to local File System. 3. Execute the HDFS commands for creating some sample directory in HDFS and then removing that directory

Unit-4	Day 9
	Detailed Schedule
1.	Hadoop Eco System **Pig:** Introduction to PIG, Execution Modes of Pig, Comparison of Pig with Databases, Grunt, Pig Latin, Data Processing operators.
Essential Assignment	
1.	Working with Pig Operators/Functions (LOAD, DUMP, FOREACH, GROUP, DISTINCT,LIMIT, ORDER BY) Write a pig script to load and store "Student data". (Student file contain Roll no, Name, Marks and GPA) a. Filter all the students who are having GPA>5 b. Display the name of all Students in Uppercase. c. Group tuples of students based on their GPA. d. Remove duplicates tuple of Student list. e. Display first three tuples from "student" relation. f. Display the names of students in ascending order.
	• 1,DDLJ,1986,3.2,7560 • 2,Xyz,1985,3.8,6300 • 3,ABC,1988,4.1,7802 • 4,PQR,1993,3.7,6022 • 5,AAA,1991,3.4,5420 • 6,ZZY,2004,3.9,4904 • 7,De danadan,1987,3.4,5623 • 8,GCET,1987,3.4,7563 • 9,PPP,1990,3.2,6244 • 10,PQQQ,2004,3.1,6956 Write a pig script to load and store movies_data.csv. **Filter:** 1. Filter movie whose rating is higher than 3.5.

	2. Store the results data from pig into new name my_movies. 3. Display all movie information the result. 4. List the movies that were released between 1950 and 1960. 5. List the movies that start with the Alphabet D. 6. List the movies that have duration greater than 2 hours. 7. List the movie names its duration in minutes. **Group Statement in PIG**. 8. List the years and the number of movies released each year. **Order by in PIG Statement.** 9. List all the movies in the ascending order of year. 10. List all the movies in the descending order of year. **Limit operator in pig.** 11. Display Top 5 movies.
Desirable Assignment	
1.	Load the file menu.csv (Category, Name, Price) and write one Pig script a. Which meals cost more than 30.00? b. Which meals contain the word “Panner”? c. Which are the 10 most expensive meals? d. For every day, what’s the average price for a meal? e. For every day, what’s the most expensive meal?

Unit-4	Day 10
.	**Detailed Schedule**
1.	Working with Pig Operators/Functions (JOIN, UNION, SPLIT, SAMPLE, AVG, MAX, COUNT, TUPLE) MAP,PIGGY BANK, PARAMETER SUBSTITUTION, DESCRIBE
Essential Assignment	
1.	Pig exercise continue… JOIN, UNION, SPLIT, SAMPLE, AVG, MAX, COUNT, TUPLE, a. Join two relations namely Student and department (Rno, DeptNo, DeptName) based on the values contain in the roll no column. b. Merge content of two relation Student and department. c. Partition a relation based on the GPA's acquired by students. d. To calculate the average marks for each student. e. Calculate maximum marks of each student. f. Count the number of tuples in a bag.
2	Assume that we have two files namely **customers.txt** and **orders.txt** in the **/pig_data/** directory of HDFS as shown below. **customers.txt** 1,Suresh,32,Ahmedabad,2000.00 2,Khivan,25,Delhi,1500.00 3,kaushal,23,Kota,2000.00 4,Chitra,25,Mumbai,6500.00 5,Hardik,27,Bhopal,8500.00 6,Sparsh,22,MP,4500.00 7,Miran,24,Indore,10000.00 **orders.txt** 102,2009-10-08 00:00:00,3,3000 100,2009-10-08 00:00:00,3,1500

	101,2009-11-20 00:00:00,2,1560 103,2008-05-20 00:00:00,4,2060 Perform JOIN on following two table • Self-join • Inner-join • Outer-join − left join, right join, and full join **Union:** merge the contents of these two relations using the **UNION** operator as shown below **Split:** Split the relation into two, one listing the employees of age less than 23, and the other listing the employees having the age between 22 and 25. **Filter:** Use the Filter operator to get the details of the students who belong to the city Chennai. **Distinct:** remove the redundant (duplicate) tuples from the relation named **student_details** **FOREACH** operator: get the id, age, and city values of each student from the relation **student_details** and store it into another relation named **student_data** **Limit:** let's sort the relation in descending order based on the age of the student and store it into another relation named **limit_data**
Desirable assignment	
1.	Write a pig script to spilt customers for reward program based on their life time values. If Life time values is >1000 and < =2000 then Silver Program If Life time values is >20000 then Gold Program

Customers	Lifetime value
Jack	25000
Smith	8000
David	12000
John	15000
Scott	12000
Lucy	28000
Ajay	12000
Vinay	30000
Joseph	21000
Joshi	25000

Unit-4	Day 11
	Detailed Schedule
1.	Working with Pig Operators/Functions: String function, date-time function, math function
Essential Assignment	
1.	**PIG :** **Eval Functions :** AVG(), BagToString(), CONCAT(), COUNT() ,COUNT_STAR() ,DIFF() , IsEmpty() ,MAX() ,MIN() ,PluckTuple() ,SIZE() SUBTRACT() ,SUM(), TOKENIZE() **String Functions :** ENDSWITH(), STARTSWITH(),SUBSTRING(), EqualsIgnoreCase(), INDEXOF(), LAST_INDEX_OF() LCFIRST(), UCFIRST(), UPPER() ,LOWER() ,REPLACE() ,STRSPLIT(), STRSPLITTOBAG(), TRIM() ,LTRIM(), RTRIM()
Desirable Assignment	
1.	Practice **Date Function** • ToDate(milliseconds) • CurrentTime() • GetDay(datetime) • GetHour(datetime) • GetMilliSecond(datetime) • GetMinute(datetime) • GetMonth(datetime) • GetSecond(datetime)

	• GetWeek(datetime) • GetWeekYear(datetime) • GetYear(datetime) • AddDuration(datetime, duration) • SubtractDuration(datetime, duration) • DaysBetween(datetime1, datetime2) • HoursBetween(datetime1, datetime2) • MilliSecondsBetween(datetime1, datetime2) • MinutesBetween(datetime1, datetime2) • MonthsBetween(datetime1, datetime2) • SecondsBetween(datetime1, datetime2) • WeeksBetween(datetime1, datetime2) • YearsBetween(datetime1, datetime2)

Unit-4	Day 12
.	**Detailed Schedule**
	Overview of Hive frame work: Hive Shell, Hive Services, Hive Metastore, Comparison with Traditional Databases, HiveQL, Tables, Querying Data and User Defined Functions
Revision of full Big Data Tools syllabus	