USN :_______________ **INTERNAL ASSESSMENT TEST 2**

Sub:	Big Data Analysis	Subject Code:	BAD 601	Branch :	AI & DS
Date:	23.05.2025	Duration: 90 min	Marks : 50	Sem : VI	OBE

Sn	Answer ALL FIVE question	Marks	CO	RBT
Q1	Explain working of Hive with proper steps and diagram. Explain types of Metastore in Hive.	5 + 5	CO 3	L2
Q2	Explain what is meant by HiveQL Data Definition Language (DDL) and Data Manipulation Language (DML). Also, explain any three HiveQL DDL commands and any three HiveQL DML commands along with their syntax and examples.	5 + 5	CO 3	L2
Q3	You are provided with a dataset related to a web application that requires basic data manipulation. (a) Demonstrate the basic CRUD operations (Create, Read, Update, Delete) in MongoDB using proper examples. (b) Explain how to create a database and drop a database in MongoDB with syntax and examples.	6 + 4	CO 4	L3
Q4	Illustrate the Spark five-layered architecture in detail with diagram. Also list the features of spark. What are the advantages of using Apache Spark over Hadoop?	5 + 3 + 2	CO 3	L3
Q5	Elaborate on the concept of Text Mining. Explain the text mining process phases with proper diagram. How is text classification achieved using the Support Vector Machine (SVM)?	2 + 5 + 3	CO 5	L2
Q6	Elaborate on the concept of Web mining. What are the web content minning technique. How does the Naive Bayes algorithm assist in classifying text data?	2 + 3 + 5	CO 5	L3

CI **CCI** **HOD**

IAT-2 Answer Key

Q1. Explain working of Hive with proper steps and diagram. Explain types of Metastore in Hive.

Ans. Hive is the data warehousing tool and also a data store on the top of the Hadoop. Hive is not a relational datbase like SQL. Hive processes structured data and integrates data from multiple hetrogemeous sources. Aditionally, also manages the constantly growing volumes of data.

Hive integrates with HDFS and Map-reduce.

Data flow and sequences and workflow steps.

Step 1: Execute Query: Hive interface (CLI or web interface) sends a query to database driver to execute the query.

Step 2: Get Plan: Driver sends the query to query compiler that parses the query to check the syntax and query plan or the requirement of the query.

Step 3: Get Metastore: Metastore sends metadata request to metastore.

Step 4: Send Metadata: Metastore sends metadata as a response to compiler.

Step 5: Send Plan: Compiler Checks the requirement and records resends the plan to driver. The parsing and compiling of the queryis complete at this place.

Step 6: Execute plan: Driver sends the execute plan to execution engine.

Step 7: Execute job: Internally, the process of execution job is a MapReduce job. The execution engine sends the job to Job tracker, which is in Name Node and it assigns this job to the tast tracker, which is in data node. Then the query executes the job.

Step 8: Metadata operations: Meanwhile the execution engine can execute the metadata operations with metastore.

Step 9: Fetch results: Execution engine receives the results from Data nodes.

Step 10: Send results: Execution engine sends the result to Driver.

Step 11: Send results: Driver sends the result to Hive interfaces.

Embedded Metastore: In Apache Hive, an embedded metastore is a configuration where the Hive metastore service runs within the same Java Virtual Machine (JVM) as the Hive service itself. This setup uses an embedded relational database, typically Derby to store metadata.

Local Metastore: In Apache Hive, a local metastore is a configuration where the metastore service runs within the same JVM as the Hive client. It connects to a database running in a separate process, either on the same machine or a remote machine. This contrasts with a remote metastore, which runs in its own JVM, separate from Hive. The Hive metastore stores metadata about Hive tables and Partitions in a relational database. In a local setup, the metastore service acts as a library within the Hive Client. Each Hive Client connects to the Database and queries it for metadata.

Remote Metastore: In Apache Hive, a remote metastore refers to a setup where the Hive metastore service runs in its own separate JVM process, distinct from the Hive server process. This configuration allows for better availability, scalability and mangeability/security, particularly when the database tier firewalled. The metstore service communicates with the hive server and the other clients via the thrift protocol.

Q2. Explain what is meant by HiveQL Data Definition Language (DDL) and Data Manipulation Language (DML). Also, explain any three HiveQL DDL commands and any three HiveQL DML commands along with their syntax and examples.

Ans: Hive Query language (HiveQL) is for querying the large datsets whivh reside in HDFS environment. HiveQl script commnads unable data defination, data manipulation and querying processing. HiveQL supports a large base of SQL users who are acquinted with SQL to extract informationfrom data warehouses.

HiveQL is similar ton SQL nfor qquerying on schema informaation at the metastore. It is one of the replacementsn of traditional approach for Map-reduce program. Instead of writting the map-reduce program in Java, we can write the query for Map-reduce and process it.

The bridge between HiveSQL process engine and Map-reduce is Hive execution engine. Execution engine processes the query and generates results same as map-reduce results.

HiveQL database copmmands for DDL dor datbase and tables are CREATE DATABASE, SHOW DATABASE, CREATE SCHEMA, CREATE TABLE. Following are the HiveQL commands:

CREATE [Db-name][EXTERNAL] TABLE[IF NOT EXISTS][<database name>]<table name>[(<column name><data type>[COMMENT<column comment>],....)]
[COMMENT <table comment>]
[ROW FORMAT <row format>]
[STORED AS <file format>]

HiveQL commands for DML are USE<database name>, DROP DATABASE, DROP SCHEMA, ALTER TABLE, DROP TABLE, and LOAD DATA.

The following is a HiveQL command:

LOAD DATA [LOCAL] INPATH '<file path>' [OVERWRITE] INTO TABLE <table name> [PARTITION (partcol1=val1, partcol=vol2......)]

Q3. You are provided with a dataset related to a web application that requires basic data manipulation.

(a) Demonstrate the basic CRUD operations (Create, Read, Update, Delete) in MongoDB using proper examples.

(b) Explain how to create a database and drop a database in MongoDB with syntax and examples.

Ans. MongoDB uses a document-oriented approach and stores data in collections (similar to tables) as documents (similar to rows), in JSON-like format (BSON).

Let's assume you are working with a collection named users inside a database named webAppDB.

(a)1. Create (Insert Data)

```
// Insert a single document
db.users.insertOne({
  name: "Alice",
  email: "alice@example.com",
  age: 28
})
// Insert multiple documents
db.users.insertMany([
  { name: "Bob", email: "bob@example.com", age: 32 },
  { name: "Carol", email: "carol@example.com", age: 25 }
])
```

2. Read (Retrieve Data)

```
// Find all documents
db.users.find()
// Find with condition
db.users.find({ age: { $gt: 25 } })
// Find one specific document
db.users.findOne({ name: "Alice" })
// Find with projection (only name and email)
db.users.find({}, { name: 1, email: 1, _id: 0 })
```

3. Update (Modify Existing Data)

```
// Update one document
db.users.updateOne(
  { name: "Alice" },
  { $set: { age: 29 } }
)
// Update multiple documents
db.users.updateMany(
  { age: { $gt: 30 } },
  { $set: { status: "senior" } }
)
```

4. Delete (Remove Data)

```
// Delete one document
db.users.deleteOne({ name: "Carol" })

// Delete multiple documents
db.users.deleteMany({ age: { $lt: 30 } })
```

(b)1. Create a Database

In MongoDB, a database is created automatically when you first store data in it.

```
// Switch to a new database
use webAppDB
// Create a collection to create the DB
db.users.insertOne({ name: "Test User", age: 22 })
```

2. Drop (Delete) a Database

```
// Switch to the database you want to drop
use webAppDB

// Drop the current database
db.dropDatabase()
```

Q4. Illustrate the Spark five-layered architecture in detail with diagram. Also list the features of spark. What are the advantages of using Apache Spark over Hadoop?

Ans. The main components of Spark stack are SQL, Streaming, R, GraphX. MLib at the applications support layer. Spark core is the processing engine. Data store provides the data to the processing engine.

I. Spark Core: All other functionality for tge spark platform is cuilt upon the universal execution engine known as spark core. It offers in-memory computations and refers to datasets in external storage sytems. APIs that are created for java, Scala, python and R make spark core accessible. These APIs levarage the complex operations of distributed processing with straightforward, high level operators.
II. Spark SQL: On the tp of Spark core, the spark SQL component adds a new data abstraction called scheme RDD that support both structure and unstructure data. Raw objects with a schema describing the data types of each column in the raw are combined to form SchemaRDD. For data queries, buisness analysts can utilize either traditional SQL or the Hive Query language. APIs are accessible in Scala, java, python and R for developers.
III. Spark streaming: It is used for analysis if data streams. This component make use of the fast schedulling technique available in the spark core. It processes RDD modification on the mini batches of data that it ingests. Spark streaming works on real time streaming analytics.
IV. Machine learning library (Mlib): It is a distribution framework for machine learning processing. According to the benchmarks, it has been tested again. Spark Mlib is 10 times faster then Apache Mahout in a Hadoop disk based vibrant. MLib includes classification, regression, clustering, collaborative filtering, dimensionality reduction and optimization primitives Mlib applies in recommendation systems, clustering and classification using Spark.
V. Spark GraphX: It is a distributed graph processing framework. It offers a graph computing API that uses the abstraction API to model user-defined graphs. Thus GraphX extends the spark RDD by introducing the Resilient Distributed Property. GraphX uses a collection of graph algorithms for programming.

The features of Spark:

I. Spark provisions for creating applications that use the complex data. In-memory Apache Spark computing engine enables up to 100 times performance with respect to Hadoop.
II. Execution engine uses both I-memory and on-disk computing. Intermediate results save in-memory and spill over to disk.
III. Provides high performance when an application accesses memory cache from disk.

IV. Contain API to define Resillient Distributed Datasets (RDDs). RDD is a programming abstraction. RDD is teh core concept in spark framework. RDD represents a collection of Objects stores distributed across many compute nodes for parallel processing. Spark stores data in RDD on different partitions.
V. Processes any kind of data, namely structured or unstructured data arriving from various sources.

Q5. Elaborate on the concept of Text Mining. Explain the text mining process phases with proper diagram. How is text classification achieved using the Support Vector Machine (SVM)?
Ans. Text minnig is the technique of analyzing collections of textual contents in order to capture key concepts themes, uncover hidden relationships, and discover the trends without requiring that you know the precise words od terms that authors have used to express those concepts.
Phase 1: Text mining process phases
Text pre-processing phases: Text clean up is the process of removing unnecessary or unwanted information. Text cleanup converts the raw data by filling up the missing values, identifies and removes outliers and resolves the consistencies.
Tokenization is a process of splitting the cleanup text into tokens, Part Of Speech (POS) tagging is a method that attempts labelling of each token with appropriate POS, word sense disambiguation is a method, which identifies the sense of words used in a sentence that gives meaning in case the word has multiple meaning, and parsing is a method, which generates a parse-tree for each sentence.
Phase 2: Features Generation
Bag of words: Method of text representation that sets collection of words and their occurances.
Stemming: Identifies the word from by its root.
Removing stop words from the feature space.
Vestor Sapce model for representing text documents as vector of identifiers, word frequencies or terms in the document index.
Phase 3: Features selection:
Dimensionality reduction, basic objective is to eliminate irrelevant and redundant data.
N-gram evaluation, finding the number of consecutive words of interest and extract them.
Noise detection and evaluation of outliers methods do the identification of unusual or suspicious items, events or observations from the data set.
Phase 4: Data minig techniques:
Applying Unsupervised and supervised learning to get the insight of the data set.
Phase 5: Analysing results:
Evaluating the outcome of the complete process.

SVM is a set of related supervised learning methods that analyze data, recognize patterns, classify text, recognize hand written characters, classify images , as well as bioinformatics and bio sequence analysis. SVM finds the best boundary (called a hyperplane) that maximizes the margin between different classes, ensuring the best separation of data points for accurate classification.

Q6. Elaborate on the concept of Web mining. What are the web content minning technique. How does the Naive Bayes algorithm assist in classifying text data?

Ans. Web mining refers to ude of techniques and algorithms that extract knowledge from the web data available in the form of web documents and services. Web mining applications are as follows:

Extracting the fragment from a web document that represents the full web document.

Identifying interesting graph patterns or pre-processing the web graph.

User identification, session creation, malicious activity detction and filtering, and extracting usage path patterns.

These are methods used to extract and analyze the contents of web pages:

1. Natural Language Processing (NLP) – Understands and interprets human language content.
2. Information Retrieval (IR) – Fetches relevant documents from large collections (e.g., search engines).
3. Text Mining – Extracts patterns and knowledge from unstructured text (e.g., topic extraction, keyword identification).
4. Wrapper Induction – Automatically extracts data from semi-structured web pages (like product listings).
5. Semantic Annotation – Adds metadata to content for better machine understanding (e.g., tagging entities).

The **Naive Bayes algorithm** is widely used for **text classification** tasks such as spam detection, sentiment analysis, and document categorization due to its simplicity and effectiveness. It is based on **Bayes' Theorem**, which calculates the probability of a class C given a set of features X (in this case, words in a document) as:

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)}$$

In practice, for text data, this means estimating the probability of a document belonging to a class (e.g., "spam") based on the frequency of words in that class. The "naive" assumption is that all words are conditionally independent given the class, allowing the likelihood P(X|C) to be computed as the product of the individual word probabilities:

$$P(C|w_1, w_2, ..., w_n) \propto P(C) \cdot \prod_{i=1}^{n} P(w_i|C)$$