USN										

Internal Assessment Test 3 – JAN 2024

Scheme of Evaluation

Sub:	**BIG DATA and ANALYTICS**					Sub Code:	**18CS72**	Branch:	**ISE**
Date:	**02/01/2024**	Duration:	**90 min**	Max Marks:	**50**	Sem/Sec:	**VII/ A, B & C**		**OBE**

Answer any FIVE FULL Questions

		MARKS	CO	RBT
1.	a. **Apply HiveQL commands for the following:** **i.Create a database named toys_companyDB and table named toys_tbl** **ii.Create a table toy_products with the following fields**	[5]	4	L3

Field	Data Type
ProductCategory	string
ProductId	Int
ProductName	String
ProductPrice	Float

Scheme and Solution: 3+2 Marks

i.Create a database named toys_companyDB and table named toys_tbl

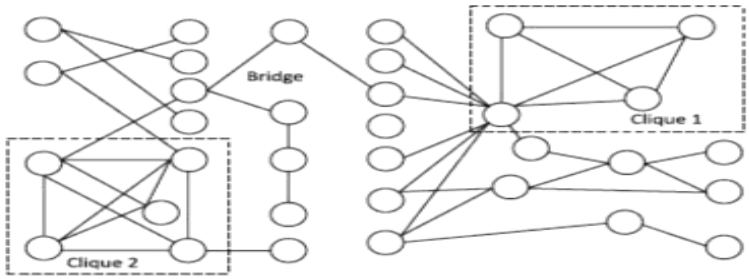
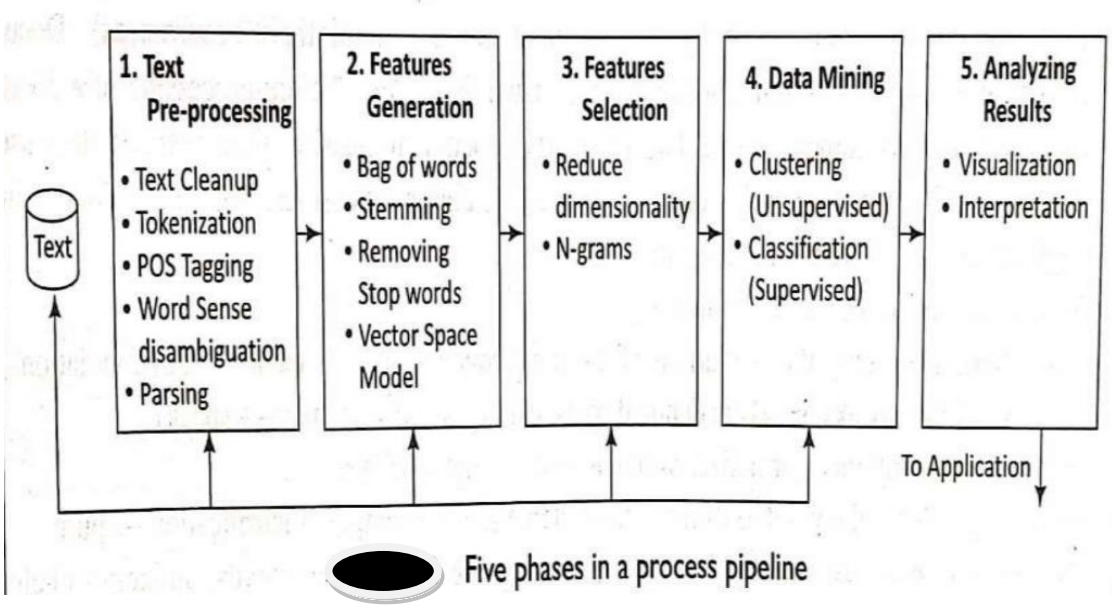
```
$HIVE_HOME/binhive – service cli
hive>set hive.cli.print.current.db=true;
hive> CREATE DATABASE toys_companyDB
hive>USE toys_companyDB
hive (toys_companyDB)> CREATE TABLE toys_tbl (
>puzzle_code STRING,
>pieces SMALLINT
>cost FLOAT);
hive (toys_company)> quit;
&ls/home/binadmin/Hive/warehouse/toys_companyDB.db
```

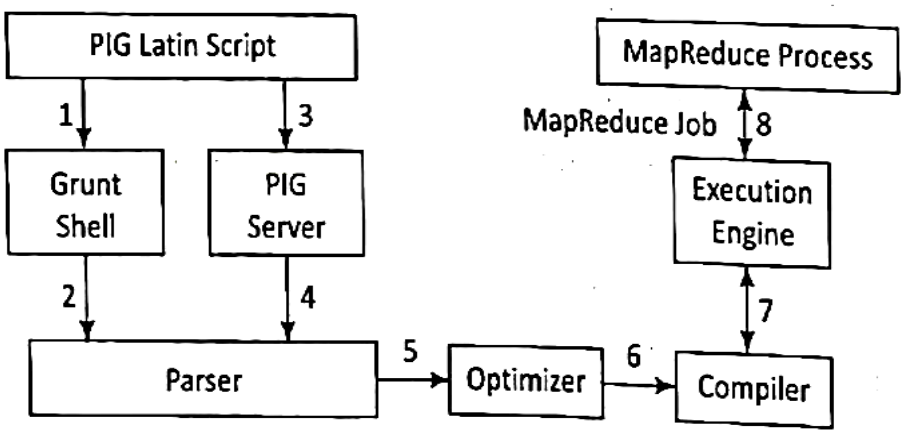
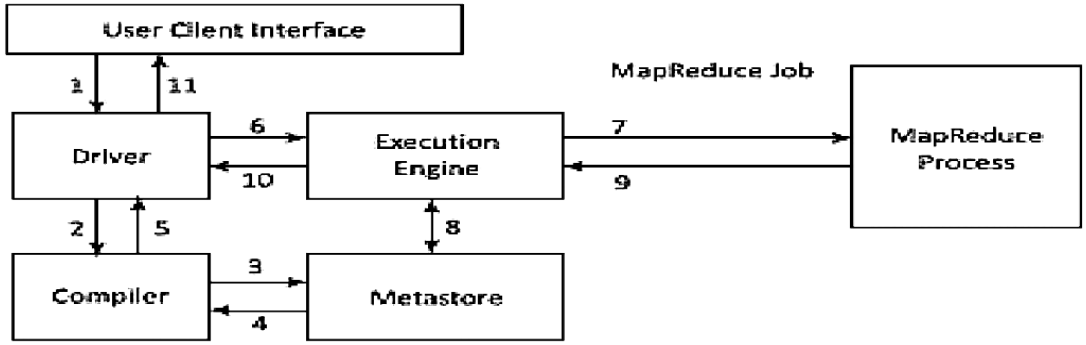
ii. Create a table toy_products with the following fields

```
CREATE TABLE IF NOT EXISTS toy_products (ProductCategory String,
ProductId int, ProductName String, ProductPrice float)
  COMMENT 'Toy details'
  ROW FORMAT DELIMITED
  FIELDS TERMINATED BY '\t'
  LINES TERMINATED BY '\n'
  STORED AS TEXTFILE;
```

	b. Apply join operations in HIVEQL by assuming the tables with attributes **toy_tbl (ProductCategory, ProductID, Product name) and ID/ProductCost data** (ProductID, DataType)
	
	Scheme and Solution: 2+1+1+1 Marks
	
	`SELECT t.ProductId, t.ProductName, p.ProductPrice`
	`FROM toy_tbl t JOIN price p`
	`ON (t.ProductId = p.Id);`
	`SELECT t.ProductId, t.ProductName, p.ProductPrice`
	`FROM toy_tbl t LEFT OUTER JOIN price p`
	`ON (t.ProductId = p.Id);`
	`SELECT t.ProductId, t.ProductName, p.ProductPrice`
	`FROM toy_tbl t FULL OUTER JOIN price p`
	`ON (t.ProductId = p.Id);`
	`SELECT t.ProductId, t.ProductName, p.ProductPrice`
	`FROM toy_tbl t RIGHT OUTER JOIN price p`
	`ON (t.ProductId = p.Id);`
		[5]		

2.	**Explain Apriori algorithm in handling big data with an example.** **Scheme: Explanation+Diagram-7+3 marks** **Solution:** • Apriori algorithm is used for frequent itemset mining and association rule mining. • Apriori algorithm is considered as one of the most well-known association rule algorithms. The algorithm simply follows a basis that any subset of a large itemset must be a large itemset. This basis can be formally given as the Apriori principle. Apriori – Example Database TID | Items 1 | {A, C, D} 2 | {A, B, C, E} 3 | {B, E} 4 | {B, C, E} Iteration 1: Candidate 1 Itemset Itemset | Support {A} | 2 {B} | 3 {C} | 3 {E} | 3 Iteration 2: Candidate 2 Itemset Itemset | Support {A, B} | 1 {A, C}* | 2 {A, E} | 1 {B, C}* | 2 {B, E}* | 3 {C, E}* | 2 Subset of a frequent itemset is also frequent Iteration 3: Candidate 3 Itemset Itemset | Support {B, C, E}* | 2 **Figure 6.8** Apriori algorithm process for adopting the subset of frequent itemsets as a frequent itemset	[10]	5	L2

3.	**Explain Cliques discover communities from social network analysis and parameters in social graph network topological analysis using centralities and PageRank.** **Scheme:** **Cliques discover explanation with diagram + parameters explanation: 5+5 marks** **Solution:** Three metrics identify groups and communities from a social graph: 1. Cliques – A clique forms by a set of vertices when each of the vertices directly connects to every other individual vertex through the edges. Detecting the cliques leads to direct discovery of communities. 2. Structurally cohesive blocks. 3. Social circles from connections and neighbourhoods  Parameters in social graph network topological analysis using centralities and PageRank are : Degree, Closeness, Effective Closeness, Betweenness, pageRank, Contacts size, Indirect contacts, Structure Diversity	[10]	6	L2
4.	**Explain five phases in a Process pipeline for Text Mining.** **Scheme: Diagram+ Explanation of each phase-2+2+2+2+2 marks** **Solution:**  Five phases in a process pipeline	[10]	6	L2
5.	**Explain Applications and features of Apache pig in big data along with architecture.** **Scheme: Applications+Features+Architecture with explanation – 2+4+4 marks** **Solution:**	[10]	4	L2

	Applications of Pig are: • Analyzing large datasets • Executing tasks involving adhoc processing • Processing large data sources such as web logs and streaming online data • Data processing for search platforms. Pig processes different types of data Features: • Allows programmers to write fewer lines of codes. Programmers can write 200 lines of Java code in only ten lines using the Pig Latin language. • Apache Pig multi-query approach reduces the development time. • Apache pig has a rich set of datasets for performing operations like join, filter, sort, load, group, etc. • Pig Latin language is very similar to SQL. Programmers with good SQL knowledge find it easy to write Pig script. • Allows programmers to write fewer lines of codes. Programmers can write 200 lines of Java code in only ten lines using the Pig Latin language. • Apache Pig handles both structured and unstructured data analysis. Architecture:  			
6.	**a. Explain Hive dataflow sequences and Workflow steps.** **Scheme: Diagram+Explanation** - 2+3 marks **Solution:**  	[5]	4	L2

1	**Execute Query:** Hive interface (CLI or Web Interface) sends a query to Database Driver to execute the query.
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.
3	**Get Metadata:** Compiler sends metadata request to Metastore (of any database, such as MySQL).
4	**Send Metadata:** Metastore sends metadata as a response to compiler.
5	**Send Plan:** Compiler checks the requirement and resends the plan to driver. The parsing and compiling of the query is complete at this place.
6	**Execute Plan:** Driver sends the execute plan to execution engine.
7	**Execute Job:** Internally, the process of execution job is a MapReduce job. The execution engine sends the job to JobTracker, which is in Name node and it assigns this job to TaskTracker, which is in Data node. Then, the query executes the job.
8	**Metadata Operations:** Meanwhile the execution engine can execute the metadata operations with Metastore.
9	**Fetch Result:** Execution engine receives the results from Data nodes.
10	**Send Results:** Execution engine sends the result to Driver.
11	**Send Results:** Driver sends the results to Hive Interfaces.

b. Differentiate between: Pig vs SQL and Pig vs Hive.
Scheme: Differences between Pig vs SQL & Pig vs Hive – 2+3 Marks
Solution:

[5]

Pig	SQL
Pig Latin is a procedural language	A declarative language
Schema is optional, stores data without assigning a schema	Schema is mandatory
Nested relational data model	Flat relational data model
Provides limited opportunity for Query optimization	More opportunity for query optimization

Pig	Hive
Originally created at Yahoo	Originally created at Facebook
Exploits Pig Latin language	Exploits HiveQL
Pig Latin is a dataflow language	HiveQL is a query processing language
Pig Latin is a procedural language and it fits in pipeline paradigm	HiveQL is a declarative language
Handles structured, unstructured and semi-structured data	Mostly used for structured data

Faculty Signature CCI Signature HOD Signature