VIT
Vellore Institute of Technology
CHENNAI

Reg. Number:	23MCA1030

Continuous Assessment Test(CAT) – II APRIL 2024

Programme	:	MCA	Semester	:	Winter 23-24
Course Code & Course Title	:	PMCA607L & Big Data Analytics	Class Number	:	CH202324050 1412
Faculty	:	Dr.K.Madheswari	Slot	:	E2+TE2
Duration	:	90 minutes	Max. Mark	:	50

General Instructions:

- Write only your registration number on the question paper in the box provided and do not write other information.
- Use statistical tables supplied from the exam cell as necessary
- Use graph sheets supplied from the exam cell as necessary
- Only non-programmable calculator without storage is permitted

Answer all questions

Q. No	Sub Sec.	Description	Marks
1		You are tasked with designing a comprehensive schema and queries for a multi-functional application in Cassandra, utilizing various data structures and concepts. The application aims to support user management, messaging, blogging, and task tracking functionalities. Construct a set of tables along with queries to perform operations as described below: **User Management (5 marks):** Design a schema for user management with a table named users. Each user should have a user_id generated as a UUID, a username, and a set of contact_ids representing the user's contacts. Implement the schema along with appropriate data types and constraints. **Messaging (5 marks):** Develop a schema for messaging functionality with a table named messages. Each message should be uniquely identified by a message_id generated as a UUID. The table should include columns for sender_id, receiver_id, timestamp, and a map of message metadata containing key-value pairs for message attributes. Construct the schema ensuring efficient storage and retrieval of messages. **Blogging (5 marks):** Create a schema for blogging capabilities with a table named posts to store blog posts. Each post should possess a post_id	20

		generated as a UUID, along with author_id, title, content, timestamp, and a set of tags associated with the post. Design the schema to support efficient querying of posts based on authors and tags. **Task Tracking (5 marks):** Enhance the schema to incorporate task tracking features with a table named tasks. Each task should be identified by a task_id generated as a UUID and include attributes such as title, description, due_date, and a list of assigned_users. Implement the schema to facilitate efficient management and retrieval of tasks. Ensure that your schema designs are optimized for scalability, data integrity, and query performance in a distributed Cassandra environment. Additionally, provide sample queries for each functionality to demonstrate the usage and effectiveness of your schema designs.	
2		Consider a scenario where you're analyzing a vast dataset containing user interactions on a social media platform. Each record in the dataset represents a user interaction event, including the user ID and the type of interaction (e.g., likes, comments, shares). Your goal is to process this dataset using MapReduce to derive meaningful insights **Dataset:** Record 1: (User1, Like) Record 2: (User2, Comment) Record 3: (User1, Share) Record 4: (User3, Like) Record 5: (User2, Like) Record 6: (User1, Like) Record 7: (User2, Comment) Record 8: (User1, Share) Record 9: (User3, Like) Record 10: (User2, Like) ● Apply the Map function to the given dataset, transforming each user interaction record into intermediate key-value pairs. ● Illustrate how the intermediate key-value pairs generated in the Map phase are shuffled and sorted based on keys, preparing them for the Reduce phase. ● Perform the Reduce operation on the shuffled and sorted key-value pairs to derive insights from the dataset. ● Calculate relevant metrics or statistics based on the aggregated results obtained from the Reduce phase. ● Extend your analysis by incorporating more complex operations into the MapReduce workflow. Consider scenarios such as filtering out specific user interactions, identifying user engagement patterns, or detecting anomalies in user behavior. Apply additional Map and Reduce functions as necessary to address these advanced analysis requirements.	10

3		You are tasked with writing a Hadoop MapReduce program to perform an inner join operation between two tables based on the "Student ID" field. The join must satisfy two filtering conditions simultaneously: a) The year of birth must be greater than (>) 1990. b) The score of course 1 must be greater than (>) 80, and the score of course 2 must be no more than (<=) 95. **Sample Data:** **Student Table:** **Student ID** **Name** **Year of Birth** 20170126453 Kristalee Copperwaite 2000 20170433596 Roeberta Naden 1997 **Score Table:** **Student ID** **score(course1)** **Score (course2)** **Score(course3)** 20170126453 93 97 80 20170140241 86 85 87 20170433596 82 60 80 ● Design a Hadoop MapReduce program to perform the inner join operation between the Student and Score tables. ● Filter records where the year of birth is greater than 1990. ● Filter records where the score of course 1 is greater than 80 and the score of course 2 is no more than 95. ● Utilize the provided sample data for both the Student and Score tables as input to the MapReduce program. ● Generate the join result, including the Student ID, Name, Year of Birth, and scores for courses 1, 2,	10
4		Let's consider a sample table named "users" with the following schema: CREATE TABLE users (user_id UUID PRIMARY KEY, username TEXT, email TEXT, age INT); ● Create a sample keyspace and table in Cassandra to use for performing CRUD operations. ● Define the schema for the table, including primary key and any secondary indexes required for querying. ● Write functions in your Cassandra program to implement the following CRUD operations: Create: Insert new records into the table. Read: Retrieve records based on primary key or secondary Index. Update: Modify existing records with new data.	10

		Delete: Remove records from the table. • Test the Cassandra program with different datasets, covering various scenarios and edge cases. • Verify the results of each CRUD operation to ensure that data is inserted, retrieved, updated, and deleted correctly. • Validate the program's behavior when handling corner cases, such as missing records or invalid input parameters.	

**********All the best **********