VIT UNIVERSITY
SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Fall Semester 2025-26

BCSE406L NoSQL Databases

CAT 1 QUESTION BANK
SLOT: A1+ TA1 and A2 + TA2

Modules covered – Module 1,2,3, 4(till Separating Data with Shard Keys)

Module 1:

1. An airline booking system initially built on a traditional RDBMS struggles with managing fluctuating schema changes for dynamic pricing, loyalty tiers, and new travel bundles. You are hired to re-architect it using NoSQL.

2. Critically analyze how schema evolution and impedance mismatch are limiting current RDBMS performance. Suggest which NoSQL model(s) to use and why.

3. A fintech company needs strong consistency for money transfers, but also requires high regional availability across the globe.

4. Using your knowledge of the CAP theorem and BASE/ACID models, explain how each requirement imposes a trade-off. Would you opt for CP, CA, or AP in this context — and where would you allow eventual consistency?

5. A news platform uses a distributed SQL database but struggles with growing read traffic in certain regions.

6. Propose a horizontal scaling solution that leverages NoSQL design patterns. How would you handle data consistency, and which type of eventual consistency is best suited for real-time updates like live comments?

7. You are working as a database administrator for a logistics company that is digitizing all of its operations. The company plans to implement a relational database management system (RDBMS) to store and manage data related to shipments, warehouses, delivery schedules, and customer information. To ensure smooth functioning, the development team must first set up the foundational components of the RDBMS. Explain in detail about the minimal requirements required for the design of RDBMS.

8. A fast-growing e-commerce startup is initially using a traditional Relational Database Management System (RDBMS) to manage its inventory, customer data, and order history. As the platform scales and user traffic increases, the system starts to face performance issues, especially during large-scale sales events. The team experiences challenges with schema changes, horizontal scaling, and handling unstructured product reviews and multimedia content. Identify at least two limitations that are likely causing bottlenecks in this scenario. Explain how NoSQL databases could resolve these issues.

9. In Database design illustrate which would benefit more from ACID compliance and which would align better with BASE principles

10. A tech company is building a suite of applications for different clients. First app is a **real-time ride-sharing app** that needs to store and retrieve user sessions and driver availability with very low latency. And the second app is a**n online retail platform** that manages millions of diverse product listings with flexible attributes like size, color, material, etc. Design a suitable NoSQL database for the same.

11. Document databases and column family databases both provide more structure than simple key-value stores. Explain the fundamental difference in their data modeling philosophy and how this distinction makes each type suitable for different kinds of applications.

12. Compare and Contrast between SQL and NoSQL Databases. Clearly Mention all the Criteria's based on which differentiation can be made.

13. How does the transactional consistency that makes a relational database strong for financial applications become a performance bottleneck for large-scale web applications, thereby justifying the adoption of NoSQL?

14. Differentiate between ACID and BASE models in database systems. In what scenarios is BASE preferred over ACID, and why?

Module 2:

15. The database developer can handle rapid reads and writes, simple lookups, and massive concurrent access, with minimal delay. Describe how features of key value database would meet this requirement.

16. Illustrate in detail about the key value data modelling term used in Hospital Database management system

17. Which key-value **design patterns** would you apply for each of the following requirements in a flash e-sale ecommerce platform?
    a. Auto-expiry of flash sale offers
    b. Fast retrieval of total review scores for products
    c. Preventing partial updates during order creation
    d. Paginated access to user order history

18. Perform Key value Database queries for the following
    a. Store and retrieve a user’s first name
    b. Retrieve the user's name

c. Add tasks to a user's to-do list and retrieve them.
d. Retrieve all tasks
e. Store and access product details.
f. Track unique visitors to a page.

19. You're tasked with implementing Redis for a fast-paced gaming platform to store user sessions, in-game state, and matchmaking queues. Design a key structure and value model that avoids collisions, supports quick lookups, and enables auto-expiry of inactive users.

20. Explain the evolution from array-based data storage to key-value databases. In your answer, discuss how the limitations of array structures in terms of scalability, data retrieval, and flexibility led to the adoption of key-value models

21. Explain how you'd prevent overwrite issues, and which design patterns (e.g., atomic aggregates, TTLs, indexes) you'd apply.

22. A mobile health app stores user preferences, sensor readings, and temporary alerts in a Redis-backed key-value store. Discuss the trade-offs of storing structured values as flat strings versus hashes or embedded structures. When would in-memory speed become a limitation in a Redis setup — and how would you handle persistence and memory pressure?

23. Redis is used for a live leaderboard feature that ranks players every 10 minutes. Create a data modeling strategy using sorted sets. Then explain how you'd handle write contention, expiration of old scores, and retrieval of top-N players using efficient commands.

24. A logistics company is developing a real-time package tracking system for its delivery fleet. The system must store and retrieve the current location of thousands of delivery vehicles, update them frequently, and support querying the latest status within milliseconds. Additionally, temporary delivery notes and error messages must expire automatically after a short period. How can a key-value database like **Redis** support this system? Explain data modeling updates, automatic expiry of alerts, and how Redis commands can help achieve low-latency and scalability.

25. Explain at least four common design patterns used in key-value databases. Provide examples for each pattern from real-world application scenarios.

26. A live quiz platform tracks each user's active session, current question number, and correct answers in Redis.
    Question:
    Write a Redis command to:

a. Set a session key with a TTL of 10 minutes
b. Increment the score after each correct answer
c. Add one more query to reset the session after quiz ends.

Module 3:

27. A social media platform wants to migrate its user posts and comments from SQL to MongoDB. Each post has likes, embedded comments, and user references. Design a document schema that supports high read speed and efficient updates. Also, explain how to design aggregation pipelines to find the top 3 most commented posts from each category.

28. An online learning platform stores course content, quizzes, and user progress in MongoDB.
    Design a collection structure that avoids over-embedding and enables analytics like "average quiz score per course per region." Write an aggregation pipeline query to get that analysis.

29. A retail app uses MongoDB for its order management system. One report requires calculating the total sales per product category per week. Discuss the aggregation design (group, unwind, project) required to build this report efficiently. What potential limitations exist if the data volume grows exponentially?

30. An e-learning platform stores course documents in MongoDB. Each document has embedded arrays of lessons and student progress. The management needs a report of average completion rate for each course.
31. Question:
    Write a MongoDB aggregation pipeline query that:
    a) Displays student progress
    b) Groups by course name
    c) Calculates the average completion percentage
    Then, rewrite the pipeline to only include courses with more than 50 students.

32. Explain in detail about the installation and execution of Mongodb and CRUD operations

33. Consider that you are working for an online store. Each order document in the orders collection contains fields like customerId, items, totalAmount, and orderDate. Using MongoDB aggregation, how would you:
    a. Find the total sales (sum of totalAmount) made in the month of July 2025? ($match, $group)
    b. Get the top 5 customers based on total amount spent? ($group, $sort, $limit)
    c. Display only the customer name and total amount spent in the result? ($project)

34. Describe in detail about the basic mongodb aggregation comments with examples

35. A startup is building an online food delivery platform. They decide to use MongoDB to manage data for restaurants, customers, orders, and menu items. Each order must store customer details, a list of food items with quantity and price and delivery status. They also need to perform analytics like total sales per restaurant, frequent customers etc. As a database engineer, explain how you would use MongoDB features to design and manage this application.

Module 4:

36. A video streaming app stores user profiles, video watch history, and region-based content restrictions. Propose a shard key strategy to enable effective horizontal partitioning. What issues might arise if user profiles are updated very frequently from multiple regions? How would you mitigate them?

37. A cloud-based education system stores millions of student records globally. Explain how you'd combine vertical and horizontal partitioning to reduce latency and isolate heavy query loads. Which fields would you choose for each partitioning technique and why?

38. An IoT platform collects temperature sensor data across hundreds of factories. The system must write millions of rows per day and still support regional dashboards with aggregated views. Design a sharded and partitioned MongoDB or NoSQL setup to achieve this. How would you prevent performance bottlenecks during peak write hours?

39. How horizontal partitioning can help with the development of application that contains 100 million users. What challenges might arise if the data is unevenly distributed among shards?

40. A hospital management system maintains a large patients table with more than 100 fields, including personal info, medical history, lab results, billing info, and insurance details. Doctors frequently query only the patient's name, ID, and latest medical records during consultations.

41. How can vertical partitioning optimize performance for doctor-related queries?

42. Explain with suitable example that how shard keys are important in partitioning. How would poor shard key selection affect the performance of database?
