SOMAIYA VIDYAVIHAR UNIVERSITY

Trim: June – Nov24 **Maximum Marks: 50 Examination: ETE Exam Date: 27/11/2024 Duration: 3 hrs**		
Programme code: 09 **Programme: MCA**	**Class:** SY	**Semester/~~Trimester~~: III**
College: K. J. Somaiya Institute of Management	**Name of the department/Section/Center:**	
Course Code: 217P09C302	**Name of the Course: NOSQL Databases**	

Instructions:

1. **Attempt any five.**
2. **Execute using respective software (MongoDB, Apache Cassandra, Neo4J).**
3. **Add the screenshots of each sub question separately in a word document & convert into PDF format.**

Question No.		Max. Marks
Q1	An e-commerce application is using MongoDB to store detailed order information. Each document in the orders collection includes embedded documents for customer details, shipping information, and items ordered. Below is the structure: {"order_id": "ORD10001", "order_date": "2024-10-01", "customer": { "customer_id": "CUST001", "name": "Ria", "email": "ria@example.com", "phone": "+123456789" }, "shipping_address": { "address_line": "123 Main Street", "city": "Metropolis", "country": "Wonderland", "zip": "54321" }, "items": [{"item_id": "ITEM1001", "name": "Laptop", "quantity": 1, "price": 1500}, {"item_id": "ITEM1002", "name": "Headphones", "quantity": 2, "price": 100}], "payment_status": "Completed", "delivery_status": "Shipped" } Write MongoDB queries for the following tasks: 1. Find all orders placed by a customer with customer_id "CUST001". 2. Retrieve all orders placed in the month of October 2024. 3. Update the delivery_status of a specific order (order_id "ORD10001") to "Delivered". 4. Calculate the total revenue generated from all orders. 5. Find all orders where the payment_status is "Pending". 6. List all orders that contain an item with item_id "ITEM1001". 7. Retrieve the names of customers who have placed an order for more than 2 items. 8. Count the number of orders shipped to a specific country. 9. Delete all orders placed before 2023-01-01. 10. Insert a new order with details for a new customer and items of your choice.	10
Q2	A healthcare system stores patient visit data in a MongoDB collection called patient_visits. The collection contains fields such as: • visit_id: Unique ID for each visit • patient_id: Unique ID for each patient • doctor_id: ID of the doctor treating the patient • visit_date: Date of the patient's visit • symptoms: List of symptoms reported by the patient • diagnosis: Doctor's diagnosis • medications: List of medications prescribed (each with dose and duration) • bill_amount: Total bill for the visit Using MongoDB's Aggregation Framework, answer the following: 1. Calculate the total bill amount each patient has incurred across all visits. 2. Find the top 3 most common diagnoses across all visits. 3. Calculate the number of visits each patient has made in the year 2024. 4. Identify the most frequently prescribed medication across all visits. 5. Calculate the average bill amount per visit for all patients.	10

	6. Find patients who have been diagnosed with more than one condition during their visits. 7. Group visits by doctor and calculate the total number of visits each doctor has handled. 8. Calculate the total revenue generated per month in 2024 based on bill amounts. 9. Identify patients who have visited more than 5 times in the last 6 months. 10. For the diagnosis of "flu", find the most common medications prescribed.	
Q3	In a retail system, customer purchase history is stored in a Cassandra database. The structure of the customer purchases table is as follows: • customer_id (UUID): Unique ID for each customer. • customer_name (TEXT): Name of the customer. • purchase_details (MAP): Maps product IDs to their respective prices. • purchase_date (SET): Stores multiple purchase dates. • categories (SET): Stores the categories of products purchased (e.g., Electronics, Clothing, Furniture). **Table Creation and Data Insertion** 1. Create the table customer_purchases with the appropriate columns for storing customer purchase details. 2. Insert 5 records into the customer_purchases table, with varied data (use product IDs, names, categories, prices, and multiple purchase dates for each cus - tomer). **Queries** 3. Write Cassandra queries to perform the following tasks: 4. Fetch all customer purchase details from the customer_purchases table. 5. Fetch the purchase details for customers who have bought items in the "Electronics" category. 6. Fetch purchase details where the total price of a specific product (e.g., product ID P002) is less than 100. 7. Fetch the customer ID and name of customers who have purchased items in both the "Clothing" and "Footwear" categories.	10
Q4.	Create a User Defined Type (UDT) called section_details to represent the section name and the number of available copies in that section. Create a table books to store the following details: • isbn (TEXT): The unique book identifier. • title (TEXT): The title of the book. • author (TEXT): The author of the book. • available_sections (LIST): List of sections where the book is available, using the section_details UDT. Write Cassandra queries to perform the following tasks: 1. Insert 3 book records into the books table with varied details (use ISBNs, titles, authors, and different section names and available copies). 2. Append new section details for a book, adding the section "Technology" with 5 available copies. 3. Update the number of copies in the "Fiction" section by increasing the available copies by 2 for a specific book. 4. Delete the section "Science" from the list of available sections for a specific book. 5. Prepend a section called "New Arrivals" with 10 copies to the list for a specific book.	10
Q5	In a Smart Home IoT System, users interact with different smart devices (like thermostats, lights, security cameras) in their homes. Each device can have interactions such as being controlled by users, monitored by users, and can send alerts to users. The system also supports scheduling actions for devices and sharing access with other users. **Graph Data Model** Create the following entities (nodes): USER: Represents a user with fields id (unique identifier), name (user's name), and email (user's email). DEVICE: Represents a smart device with fields id (unique identifier), type (e.g., thermostat, light, security camera), and status (e.g., on/off). Create the following relationships: • USER -> CONTROLS -> DEVICE - Represents a user controlling a device. • USER -> MONITORS -> DEVICE - Represents a user monitoring a device. • USER -> RECEIVES_ALERT -> DEVICE - Represents a user receiving alerts from a device. • USER -> SHARES_ACCESS ->USER - Represents one user sharing access to a device with another user. • DEVICE -> SCHEDULES_ACTION -> USER- Represents a device scheduling actions for a user. Write the following Neo4j queries: 1. Insert 3 user records with different details and 2 device records. 2. Insert relationships for user actions. 3. Fetch all devices controlled by a specific user, including the status of each device. 4. Find all users who have shared access to a specific device. 5. Get all users who have scheduled actions for a specific device.	10

Q6	Wow foods provide personalized experience in purchasing the food materials to each customer. Identify the design decisions to be made and challenges in designing the database for Wow foods. Based on your analysis of this scenario, create a MongoDB database to fetch necessary information.	10