

# CMM702 Advanced Databases

Academic Year	2025/2026
Semester	2
Module Number	CMM702
Module Title	Advanced Databases
Assessment Method	Coursework
Deadline (time and date)	16th April 2026 - 11.59 pm IST
Submission	Assessment Dropbox in the Module Study Area in CampusMoodle. <b>Attending CW VIVA is compulsory.</b> <b>Failing to do so will set you CW grade to a "Fail"</b>
Word Limit (see Assessment Word Limit Statement)	N/A
Module Co-ordinator	Yamuna Dulanjani Indrajith Ekanayake

What knowledge and/or skills will I develop by undertaking the assessment?

To provide a systematic understanding of up-to-date issues, techniques, and technologies for developing robust, usable, and scalable database management systems for big data.

**On successful completion of the assessment, students will be able to achieve the following**

**Learning Outcomes:**

On completion of this module, students are expected to be able to:

1. Critically appraise relational database principles and practices in the context of transferability to complex, large-scale data stores.
2. Develop robust and scalable systems that integrate web technologies for databases and database connectivity.
3. Critically appraise relational and schema-less database offerings (e.g., NoSQL) for a significant technical problem.
4. Design and implement a solution to a significant industry-focused problem, providing insights and conclusions about challenges, opportunities, and risks for big data management.

**Please also refer to the Module Descriptor, available from the module Moodle study area.**

What is expected of me in this assessment?

**Part A**

**Section 1**

Questions 1 through 3 are based on the following case study. Carefully read through and answer the questions.

## What is expected of me in this assessment?

MediSphere Healthcare Services (MHS) is a large-scale healthcare provider operating across multiple states in the USA. It delivers healthcare services through three main channels: Hospitals & Clinics, Telemedicine Platforms, and Corporate Wellness Programs. Hospitals & Clinics provide in-person consultations, treatments, and surgeries. Telemedicine Platforms enable online consultations, prescriptions, and follow-up services via a website and mobile app. Corporate Wellness Programs are designed for organizations to improve employee health and productivity through medical check-ups, fitness sessions, and awareness workshops.

MHS has over 100 healthcare facilities across various states to ensure wide patient coverage. Employees of MHS can be categorized as Doctors, Nurses, Pharmacists, IT Specialists, Laboratory Technicians, and Medical Administrators. In addition, administrative staff such as Receptionists, Accountants, HR Officers, Insurance Coordinators, and Marketing Executives support daily operations. Typically, an employee is assigned to a single facility, but some administrative staff oversees multiple facilities. Each employee is uniquely identified by an Employee ID. The MHS information system stores employee details such as name, designation, date of joining, date of termination, salary, email, phone number, and home address. The system also tracks the assignment history of employees, including start and end dates of their postings at specific facilities.

Each healthcare facility is uniquely identified by a Facility ID, along with details such as facility name, address, postal code, and contact number. Each facility has a Facility Manager responsible for daily operations, ensuring patient care quality, compliance with healthcare standards, and staff performance. Doctors and Nurses are classified based on their specializations, such as Cardiology, Pediatrics, Neurology, Orthopedics, General Medicine, and Emergency Care.

MHS operates four main types of facilities: Hospitals, Clinics, Telemedicine Hubs, and Diagnostic Centers. Each facility has a unique ID, capacity, location, and year of establishment. Except for telemedicine hubs, which provide remote healthcare, all other facilities are registered with the respective State Health Department. Each facility is also responsible for serving designated health regions, uniquely identified by a Region Code. The MHS system tracks the coverage area of each facility and provides real-time service availability updates through the MHS mobile app.

Facilities typically operate in three shifts: morning, afternoon, and night. While Doctors and Nurses can be assigned across Hospitals, Clinics, and Diagnostic Centers, Laboratory Technicians primarily work in Diagnostic Centers. Pharmacists manage prescriptions at Hospitals and Clinics, whereas Medical Administrators coordinate patient records and insurance claims across all facilities. IT Specialists maintain the systems that manage patient data, appointments, insurance claims, and treatment histories.

There are two types of IT tasks. Scheduled Maintenance (system upgrades, compliance checks, backups, security patches). On-Demand Issue Resolution requested by staff at facilities. Each IT task is uniquely identified by a Ticket ID, with details such as task type, start date, end date, and assigned IT personnel. For on-demand issues, the system additionally stores the requested by, request date, resolution date, and resolution notes. IT Specialists decide whether to accept an on-demand request based on team workload and urgency.

MHS also partners with pharmaceutical suppliers, medical equipment providers, and insurance companies to manage procurement and distribution of medicines, diagnostic kits, surgical equipment, and insurance services. Each partner is uniquely identified by a Partner ID, along with the company name, the type of service/supply provided, and contact details.

## What is expected of me in this assessment?

The system tracks all patient interactions, uniquely identified by a Patient Visit ID. For each visit, the system records details such as doctor name, scheduled date, service type (consultation, surgery, diagnostic test, etc.), and visit status. During the care process, the system also captures duration, medications prescribed, diagnostic test results, insurance claims, and follow-up recommendations.

Additionally, the system tracks patient enrollments in wellness programs, medical assessments, and health certifications (such as vaccination certificates or fitness-for-work documentation).

### **Question 1 [LO2]**

Draw a complete Conceptual Model for the MHS. You need to include all the entity types, relationship types, multiplicity constraints, all types of attributes, and primary keys that you have identified based on the above description. Note that failing to add any information will lead to partial credits.

### **Question 2 [LO2]**

Based on the above Conceptual Model, build a Logical Model for MHS based on relational modeling. It is required to include all the relations, attributes, primary keys, and foreign keys that you have identified from the case study. Missing any information will end up with partial credits.

### **Question 3 [LO1]**

Based on the scale of the business and its operating across several states, discuss in detail the advantages and disadvantages of implementing it in a distributed database environment. Make sure to include at least three advantages and three disadvantages of such an approach. Furthermore, summarize three candidate commercial-level database management systems if you plan to implement it.

## **Section 2**

### **Question 1 [LO2]**

The following is a fraction of an E-Commerce database schema that tracks the relations between Customers and their Orders.

Customer ([cust\\_id](#), name, email, phone, age, gender, city, state)

Order ([order\\_id](#), order\_date, total\_amount, payment\_method, cust\_id) [foreign key to customer]

Consider that the following queries account for the majority of the workload. They are roughly equal in terms of frequency and importance.

1. List the names, emails, and cities of customers filtered by a given state and gender. Assume that customers are fairly evenly distributed across states, but gender distribution is skewed (one gender dominates).
2. List the order IDs and payment methods for a specific total amount requested by the user. More than 70% of orders have similar amounts (e.g., \$50–\$60 range).
3. Retrieve the order with the earliest order\_date in the system.
4. List all details about customers who have placed at least one order paid with “Credit Card”. (Hint: Consider the cost of joins.)

## What is expected of me in this assessment?

A workload analysis shows that read queries dominate updates, so indexes are crucial for performance. Based on this, design the minimum set of indexes required to optimize these queries:

- Decide whether to use a B+ tree or Hash indexes.
- Identify which attributes should be indexed.
- Indicate whether the indexes should be clustered or unclustered.

Justify your design choices based on the cost model (approximate reasoning based on equations is sufficient, exact values not required).

Hint: Start by expressing the above requirements as 4 SQL queries, then derive the indexing strategy.

## Question 2 [LO1]

Given that general external sorting is used, calculate the information below (questions 1 through 5) for each of the specified cases.

Case 1: A file with 4,000 pages and 5 available buffer pages

Case 2: A file with 100,000 pages and 8 available buffer pages

1. How many runs will you need to complete the 1st pass?
2. How many passes will you need to get the file completely sorted?
3. What is the total I/O cost for the sorting process?
4. How many buffer pages will you need to sort the file completely in 2 passes?
5. If the number of available buffer pages doubles in each case, how does it impact the number of passes required?
6. Plot the behavior of required number of passes (P) to completely sort a file of N pages when only 3 buffer pages are available.

## Question 3 [LO1]

Below are three relations from a Healthcare Patient Management Database of a particular state in the US that represent patients, the medical tests they have taken, and the hospitals where these tests were conducted.

Assume that the test code can be in the range of 1 – 100 and is generated uniquely across all test types, considering the department and the batch number.

- Patients(ptn, name, age, city)
- Medical\_Tests(ptn, hospital\_id, test\_code)
- Hospitals(hospital\_id, hospital\_name, state)

Assume that equal-sized fields exist in each relation and that patients are uniformly distributed across hospitals in the Medical\_Tests relation.

Assumptions related to Storage Information:

- Patients relation: 6000 pages, 100 tuples per page
- Medical\_Tests relation: 12,000 pages, 100 tuples per page

## What is expected of me in this assessment?

- Hospitals relation: 3000 pages, 100 tuples per page

Assumptions related to Indexes Available:

- Index 1: Unclustered B+ tree index on `Medical_Tests(test_code, ptn)`
- Index 2: Unclustered hash index on `Patients(ptn)`
- Index 3: Unclustered B+ tree index on `Patients(ptn, name)`

Other assumptions:

- Only 5% of the tuples satisfy the selection condition of `test_code > 80`.
- 500 buffer pages are available.
- A hash index takes 1.2 I/Os, and a B+ tree index takes 3 I/Os to find the RID of a tuple when the key is given.
- You may ignore the cost of storing RIDs in index entries for simplicity.

Consider the following query

```
SELECT p.ptn, p.name, mt.test_code
FROM Patients p, Medical_Tests mt
WHERE p.ptn = mt.ptn AND mt.test_code > 80;
```

You need to evaluate the query using Index Nested Loop Join (INLJ) and Sort-Merge Join (SMJ) and determine an efficient execution plan for both cases. Note that you can use only the above-mentioned indexes while reaching your solution.

You should cover the following criteria:

- Describe the two plans(selected efficient execution plan) individually based on the above joining methods.
- Show the cost estimations for disk I/O for each plan.

\* You may ignore the cost for storing rids in data entries of indexes in your calculations for simplicity. Write formulas and sub-formulas whenever possible to support your answer. Marks will be given for each step that is shown in the estimation.

## Part B [LO2, LO3, LO4]

### Question 1 [LO3, LO4]

Consider the following NoSQL types:

- Key/Value stores
- Document databases
- Column-Family stores
- Graph databases

For each of the above types, think about a real-world system or a component of a large-scale system where a selected NoSQL type suits the most.

- Explain in detail, critical requirements of each identified system where NoSQL features are required.

## What is expected of me in this assessment?

- B. Summarize how the above-identified critical requirements are mapped with the key characteristics of the selected NoSQL database type.
- C. Discuss the advantages and disadvantages of the particular NoSQL in the discussed application.
- D. Take one of the examples and discuss why it is beneficial to use NoSQL rather than using a relational database.

Note that your application does not have to exist but make sure to explain the concept of the application clearly.

### Documentation Instructions:

Your answer to Question 1 should include justifications and descriptions of the aspects 'A' through 'C' for all four NoSQL types individually. Your answer for each application should be based on the above parts 'A' to 'C', and each of such applications should be elaborated using at least 500 words. For the section 'D', few paragraphs with at least 150 words together would be sufficient.

## Question 2 [LO2, LO3, LO4]

For question 2, use the provided code *clicklogs.zip* folder. The zip file contains a web-based click logging system for user interface research, capturing timing data across different device types and interface variations.

Here's how it works,

- User selects device type (Android/PC)
- System randomly assigns feedback mode (shows mean duration or not)
- User taps the button up to 50 times per session
- Captures precise timing data: start/end timestamps, duration, sequence
- Runs 2 interface variations (feedback vs no-feedback)

However, if you closely inspect the *index.html*, you would notice that the *saveTaps.php* PHP server file is missing. Your work starts from here...

1. Create *saveTaps.php* to receive the POST data (It's not mandatory to use a PHP backend; any other backend would also work). The following data needs to be captured per tap:
  - a. Tap the sequence number
  - b. Start/end timestamps
  - c. Interface type (feedback/no-feedback)
  - d. Session identifier
  - e. Device platform
2. Ingest data to Firebase Firestore and store each tap record in a Firebase Firestore collection called *tap\_logs*. While ingesting, design the Firebase Firestore document structure for storing tap data. Consider:
  - a. What fields should be indexed for efficient querying?
  - b. How will you handle the relationship between session data and individual taps?
  - c. Should you store calculated fields like duration or compute them during queries?
3. Share your user interface with your colleagues and ask them to experiment by tapping from multiple devices (You can host the webpage using GitHub Pages).
4. After collecting a significant amount of test data, write MongoDB queries to answer:
  - a. Calculate the mean tap duration for Android vs PC users
  - b. Compare the average tap duration between "feedbackshown" vs "nofeedback" interfaces?

## What is expected of me in this assessment?

- c. How many users completed both interface variations vs dropped off after the first?

### Question 3 [LO4]

For Question 3, use the data provided with the 'books.json' file and load it into a MongoDB instance. Write MongoDB commands for the following requirements and provide them in the document. Provide the results you got through the commands used (as a screenshot in the document).

Note that for each requirement given below, the expectation is a single MongoDB query, and not in multiple parts. During the viva session, these queries need to be executed and explained.

1. Create a database called "iitdb" and a collection called "books" and load the given data set to the "books" collection.
2. List only the titles of all the published books.
3. Find the books with 300 to 450 pages.
4. Count the number of books where the author is 'Robi Sen'.
5. Write a query to find the books with a title that starts with the word 'Mongo' and display only the title and authors.
6. Find the books of category "Internet" according to the first element in the "categories" Array. Display only the first author (first element in the "authors" Array), and the title. Arrange the name of the first author in ascending order and for that same first author, titles should be in descending order. A single query is expected that satisfies all of the above requirements.
7. Write a query to find the books written by more than four authors and display only the title and the number of authors.
8. Write an aggregation pipeline to count the number of published books for each category. Display the number of books in the "Internet" category. A single query is expected that satisfies all the above requirements.

Documentation Instructions:

In your document, you should include MongoDB commands for each task given above and the results you obtained where appropriate.

## How will I be graded?

A grade will be provided for each criterion on the feedback grid which is specific to the assessment. The overall grade for the assessment will be calculated using the algorithm below.

<b>A</b>	At least 50% of the feedback grid to be at Grade A, at least 75% of the feedback grid to be at Grade B or better, and normally 100% of the feedback grid to be at Grade C or better.
<b>B</b>	At least 50% of the feedback grid to be at Grade B or better, at least 75% of the feedback grid to be at Grade C or better, and normally 100% of the feedback grid to be at Grade D or better.
<b>C</b>	At least 50% of the feedback grid to be at Grade C or better, and at least 75% of the feedback grid to be at Grade D or better.

## How will I be graded?

<b>D</b>	At least 50% of the feedback grid to be at Grade D or better, and at least 75% of the feedback grid to be at Grade E or better.
<b>E</b>	At least 50% of the feedback grid to be at Grade E or better.
<b>F</b>	Failing to achieve at least 50% of the feedback grid to be at Grade E or better.
<b>NS</b>	Non-submission.

## Feedback grid

GRADE	A	B	C	D	E	F
DEFINITION / CRITERIA (WEIGHTING)	EXCELLENT Outstanding Performance	COMMENDABLE/VERY GOOD Meritorious Performance	GOOD Highly Competent Performance	SATISFACTORY Competent Performance	BORDERLINE FAIL	UNSATISFACTORY Fail
<b>PART A</b> <b>Conceptual Model, Logical Model and the need for having a distributed environment.</b> <b>Weight: 3</b>	Both the conceptual model and the logical model are drawn with all the required entities and relationships. All the complex relationships have been captured and different combinations of entity types like super class sub class have been identified. Advantages and disadvantages of implementing the specified scenario in a distributed system is well identified and defended with examples	Most of the entities and relationships covered in both models are present, despite a few being missing.  Advantages and disadvantages of implementing this scenario are identified and explained with examples	Most of the entities and relationships covered in both models are present, despite a few being missing.  Discussion of implementing this system in a distributed environment is justified to some extent, although it could have been further improved.	At least one of the models drawn properly and the second one is attempted to some level.  Discussion of implementing this system in a distributed environment is justified to some extent, although it could have been further improved.	At least one model attempted out of the conceptual and the logical although there is a good number of possible improvements.  Attempted the discussion on the need for a distributed system, although quite a lot of missing components.	Either the three tasks not attempted at all or only attempted one of them in a very poor level.

GRADE	A	B	C	D	E	F
DEFINITION / CRITERIA (WEIGHTING)	EXCELLENT Outstanding Performance	COMMENDABLE/VERY GOOD Meritorious Performance	GOOD Highly Competent Performance	SATISFACTORY Competent Performance	BORDERLINE FAIL	UNSATISFACTORY Fail
<b>Part A</b>  <b>Usage of appropriate indexing strategies gives a certain scenario and justification for the selection.</b> <b>Weight: 1</b>	All the sub-questions were addressed with Appropriate SQL queries to begin with, an appropriate selection of the indexing, and justification followed by all the selections. The selection of indexes should be done considering all the queries as a whole and how indexes used in one query can impact the database as a whole	All SQL queries, index selection and justification are good but the selection if index is not done considering how one index selection can impact the other, basically isolated selection of indexes is made.	SQL queries and index selection are done well, but the level of justification is moderate. Possibly an isolated selection of indexes without considering all of them as a whole	SQL queries and index selection are done, but the justification is poor. The indexes are selected without considering all of them as a whole database	SQL queries and index selection are there, but no justification for the selection.	Only the SQL queries are there. No index selection, justification.
<b>Part A</b>  <b>For a given sorting algorithm, calculate resource requirements for different scenarios.</b> <b>Weight: 1</b>	For general external sorting, for the two cases provided, the number of runs, the number of passes, the costs, and the pattern of them are identified and drawn as a graph. Details of all the calculations and a justification behind the graph are clearly mentioned	At least 3 out of 4 components of the questions were answered well with justifications and details. The graph is complete with all the necessary justification.	At least 3 out of 4 components of the questions were answered well with justifications and details. The graph is done without the justifications.	At least 3 out of 4 components of the questions were answered well with justifications and details. Graph is attempted, but there are possible improvements.	At least 2 out of 4 components of the questions were answered well with justifications and details. Graph is not attempted.	At least 2 out of 4 components of the questions were answered well with justifications and details. The graph may not have been attempted.

GRADE	A	B	C	D	E	F
DEFINITION / CRITERIA (WEIGHTING)	EXCELLENT Outstanding Performance	COMMENDABLE/VERY GOOD Meritorious Performance	GOOD Highly Competent Performance	SATISFACTORY Competent Performance	BORDERLINE FAIL	UNSATISFACTORY Fail
<b>Part A</b> <b>Comparison of different index plans and selecting the best plan for a given scenario with available indexes</b> Weight: 2	Both Index Nested Loop Join (INLJ) and Sort Merge Join (SMJ) have been described, and cost calculations are also done clearly. Both of the components are justified with discussions.	Both index plans are discussed and calculations are there where necessary. But the justification for the process is at a moderate level.	Both index plans are discussed, but at a moderate level, and calculations are provided where necessary. But the justification for the process is at a poor level.	Both index plans are discussed, and calculations are there, but there is no justification/discussion that follow.	Only one index plan is there without a discussion/justification	Not attempted the index plans at all.
<b>PartB</b> <b>Understanding different NOSQL types and their practical applications</b> Weight: 1	Excellent understanding of all four NOSQL database types and connecting these types to practical applications.	Very good understanding of all NOSQL types with a moderate level of discussion on their practical applications.	Good understanding of all NOSQL types with a moderate level of discussion on their practical applications.	Sufficient understanding of all NOSQL types with moderate level of discussion on their practical applications.	Only two or less of the four NOSQL types understood or with poor understanding of practical applications.	One or fewer NOSQL types are well understood, and there is a very poor understanding of practical understanding.

GRADE	A	B	C	D	E	F
DEFINITION / CRITERIA (WEIGHTING)	EXCELLENT Outstanding Performance	COMMENDABLE/VERY GOOD Meritorious Performance	GOOD Highly Competent Performance	SATISFACTORY Competent Performance	BORDERLINE FAIL	UNSATISFACTORY Fail
<b>Part B</b> <b>Firebase Firestore ingestion, schema design, and indexing strategy</b> <b>Weight: 1</b>	Clean document model for taps and sessions that is justified. Correct choice of embedding vs referencing, clear session linkage, durable session identifier, stored duration where useful, consistent server time standard, and well-reasoned compound indexes that match query patterns (platform, interface, session, sequence). Brief rationale on write vs read trade-offs and shard key choice (if applicable).	Sound document model with correct fields. Indexes provided for main queries, minor mismatches, or limited rationale. Session-tap relationship handled sensibly.	Usable schema with most fields present. Indexes exist, but not fully aligned to the query workload or scalability. Relationship is workable but not optimal.	The schema is ad hoc or redundant. Indexes are missing or poorly chosen. The relationship between the session and taps is unclear.	Incomplete schema, important fields or relationships missing. No meaningful indexing.	No working ingestion or schema.
<b>PART B</b> <b>Deployment, sharing, and data collection</b> <b>Weight: 1</b>	UI hosted publicly (e.g., GitHub Pages), endpoint reachable, tested across devices. Clear participant instructions, evidence of multi-device data, and a short log of collection sessions.	Public hosting and working collection with multiple test entries from at least two device types.	Hosts and collects data, but testing is limited to a single device type or a small sample.	Hosted, but the collection is unreliable or inconsistent.	Hosting incomplete, few, or no usable records.	Not hosted or not accessible.
<b>PART B</b> <b>Firebase Firestore queries and analysis outcomes</b> <b>Weight: 1</b>	Correct, efficient aggregation pipelines for all three tasks. Pipelines use correct group keys and filters, avoid unnecessary stages, and handle edge cases (incomplete sessions,	All three queries are correct with minor inefficiencies. Outputs and short notes provided.	Queries mostly correct but one contains a logic flaw or mis-grouping. Outputs present.	Queries partially correct; two or more issues in grouping or filtering, or missing one output.	Queries are largely incorrect or missing two tasks.	No working queries.

GRADE	A	B	C	D	E	F
DEFINITION / CRITERIA (WEIGHTING)	EXCELLENT Outstanding Performance	COMMENDABLE/VERY GOOD Meritorious Performance	GOOD Highly Competent Performance	SATISFACTORY Competent Performance	BORDERLINE FAIL	UNSATISFACTORY Fail
	missing fields). Presents concise numeric outputs and a brief interpretation per query.					
<b>PartB</b> <b>Hands-on with MongoDB</b> <b>Weight: 3</b>	Excellent understanding of how to use MongoDB for a given problem	Very good understanding of how to use MongoDB for a given problem	Good understanding of how to use MongoDB for a given problem	Satisfactory understanding of how to use MongoDB for a given problem	Poor understanding of how to use MongoDB for a given problem	Very poor understanding of how to use MongoDB for a given problem

***Coursework received late, without valid reason, will be regarded as a non-submission (NS) and one of your assessment opportunities will be lost.***

## What else is important to my assessment?

### What is plagiarism?

"Plagiarism is the practice of presenting the thoughts, writings or other output of another or others as original, without acknowledgement of their source(s) at the point of their use in the student's work. All materials including text, data, diagrams or other illustrations used to support a piece of work, whether from a printed publication or from electronic media, should be appropriately identified and referenced and should not normally be copied directly unless as an acknowledged quotation. Text, opinions or ideas translated into the words of the individual student should in all cases acknowledge the original source" ([RGU 2022](#)).

### What is collusion?

"Collusion is defined as two or more people working together with the intention of deceiving another. Within the academic environment this can occur when students work with others on an assignment, or part of an assignment, that is intended to be completed separately" ([RGU 2022](#)).

For further information please see [Academic Integrity](#).

### What is the Assessment Word Limit Statement?

It is important that you adhere to the Word Limit specified above. The Assessment Word Limit Statement lists what is included and excluded from the word count, along with the penalty for exceeding the upper limit.

### What if I'm unable to submit?

- The University operates a [Fit to Sit Policy](#) which means that if you undertake an assessment then you are declaring yourself well enough to do so.
- If you require an extension, you should complete and submit a [Coursework Extension Form](#). This form is available on the RGU [Student and Applicant Forms](#) page.
- Further support is available from your Course Leader.

### What additional support is available?

- [RGU Study Skills](#) provides advice and guidance on academic writing, study skills, maths and statistics and basic IT.
- [RGU Library guidance on referencing and citing](#).
- [The Inclusion Centre: Disability & Dyslexia](#).
- Your Module Coordinator, Course Leader and designated Personal Tutor can also provide support.

### What are the University rules on assessment?

The University Regulation '[A4: Assessment and Recommendations of Assessment Boards](#)' sets out important information about assessment and how it is conducted across the University.