CSCI 485 – MongoDB Project Deliverable 3

Indexing, Workload Analysis & Relationship Design

Due Date: October 21, 2025

Weight: 50 points
Submission Format: Single PDF document with accompanying .js files

---

1. Overview

Overview

Building on your **Deliverable 2 (Database Design & Collection Architecture)**, this deliverable focuses on **performance optimization and advanced MongoDB features**. Students will analyze query workloads, create and justify indexing strategies (including text and compound indexes), and document data access patterns. Optional enhancements using **GridFS** (for file storage) or **GeoJSON** (for geospatial data) should be explored if relevant to your domain.

2. Deliverable Components

A. Indexing Strategy & Justification

You must:

1. **Create indexes** for each collection used in your project (minimum 5 total indexes).
   - Include **compound**, **text**, or **2dsphere** indexes if applicable.
   - Consider partial or unique indexes where appropriate.
2. **List all indexes** in a summary table:
   - Collection name
   - Index key(s)
   - Type (single field, compound, text, geo, partial, etc.)
   - Purpose / query supported
3. Include your **index creation script** (create_indexes.js) in your submission.
4. Explain the **reasoning** behind each index:
   - What query pattern does it optimize?
   - How frequently is that query executed?
   - Why was this index type chosen?

B. Workload & Operations Analysis

Identify the most common database operations your application will perform. For each, include:

- **Operation type:** (Read, Write, Update, Aggregate, etc.)
- **Criticality:** (High, Medium, Low)
- **Estimated frequency:** (e.g., “Many per minute”, “Few per day”)
- **Targeted collection(s)**

Summarize your workload analysis in a table. Example:

Operation	Type	Criticality	Frequency	Target Collection
User Login	Read	High	High	users
List Assignments	Read	High	Medium	assignments
Submit File	Write	High	Medium	submissions

Include a short discussion on **which operations you optimized with indexes** and why.

C. Design Patterns used & Anti-Patterns avoided

Clearly explain at least **two design patterns used** (e.g., referencing pattern, embedding pattern, partial index pattern) and **two anti-patterns avoided**. Some anti patterns are:

- Over-embedding large subdocuments
- Over-indexing (too many indexes slowing down writes)
- Missing indexes on frequent queries
- Using regex without index support

D. Relationship & Schema Diagrams

Create two diagrams:

1. **ER Diagram (Entity Relationship)** showing logical entities, primary and foreign keys.
2. **Collection Relationship Diagram** showing actual MongoDB collections, embedding, and referencing decisions.

You may use **Lucidchart**, **Draw.io**, or any diagram tool. Include these diagrams in your PDF.

Label relationships clearly:

- 1:1, 1:Many, or Many:Many
- Indicate whether relationships are implemented by **embedding** or **referencing**.

E. GridFS and/or GeoJSON

If your project uses:

- **GridFS** for large file storage — explain its purpose and show example metadata schema and indexes used on fs.files.
- **GeoJSON** for spatial data — describe its purpose and demonstrate how a 2dsphere index supports queries such as find nearest.

3. Technical Guidelines

- Use **descriptive field names** (camelCase recommended)
- Include **index creation code** with clear comments
- Ensure **sample data** supports your chosen queries
- Diagrams should be readable and logically consistent
- Your document must be **clear, organized, and professional**

4. Deliverables to Submit

Your submission folder must contain:

1. Deliverable3_Report.pdf – written explanation, tables, and diagrams
2. create_indexes.js – index creation scripts with comments
3. (Optional) sample_queries.js – example queries showing index use. Ignore if already done

5. Submission Notes

- Submit your PDF and scripts through **VIULearn** before the deadline.
- Late submissions will follow the standard course late policy.
- You may reuse and refine materials from Deliverable 2.