Declaration

I, Yijiang Liu, declare that this assignment, titled A3, is my own original work and has not been copied from any other source except where explicitly acknowledged. I have not engaged in plagiarism, collusion, or any other form of academic misconduct in the preparation and submission of this assignment. All sources of information and data used in this assignment have been properly cited and referenced in accordance with the prescribed guidelines. I have not used unauthorized assistance in the preparation of this assignment and have not allowed any other student to copy my work. I am aware that any breach of academic integrity may result in disciplinary action as per the policies of Monash University, which may include failing this assignment or the course, and further academic penalties.

Signature: Yijiang Liu Date: 10/07/2024

Self-Evaluation

Rate your performance for each criteria. Put a ✓(tick) in the box where you think your work belongs.

Criteria	Exceeds Expectations	Meets Expectations	Needs Improvement	Fail to meet expectations
Introduction		✓		
Main Body		✓		
Reflection		✓		
AI Acknowledgement		✓		

Section 1: Introduction

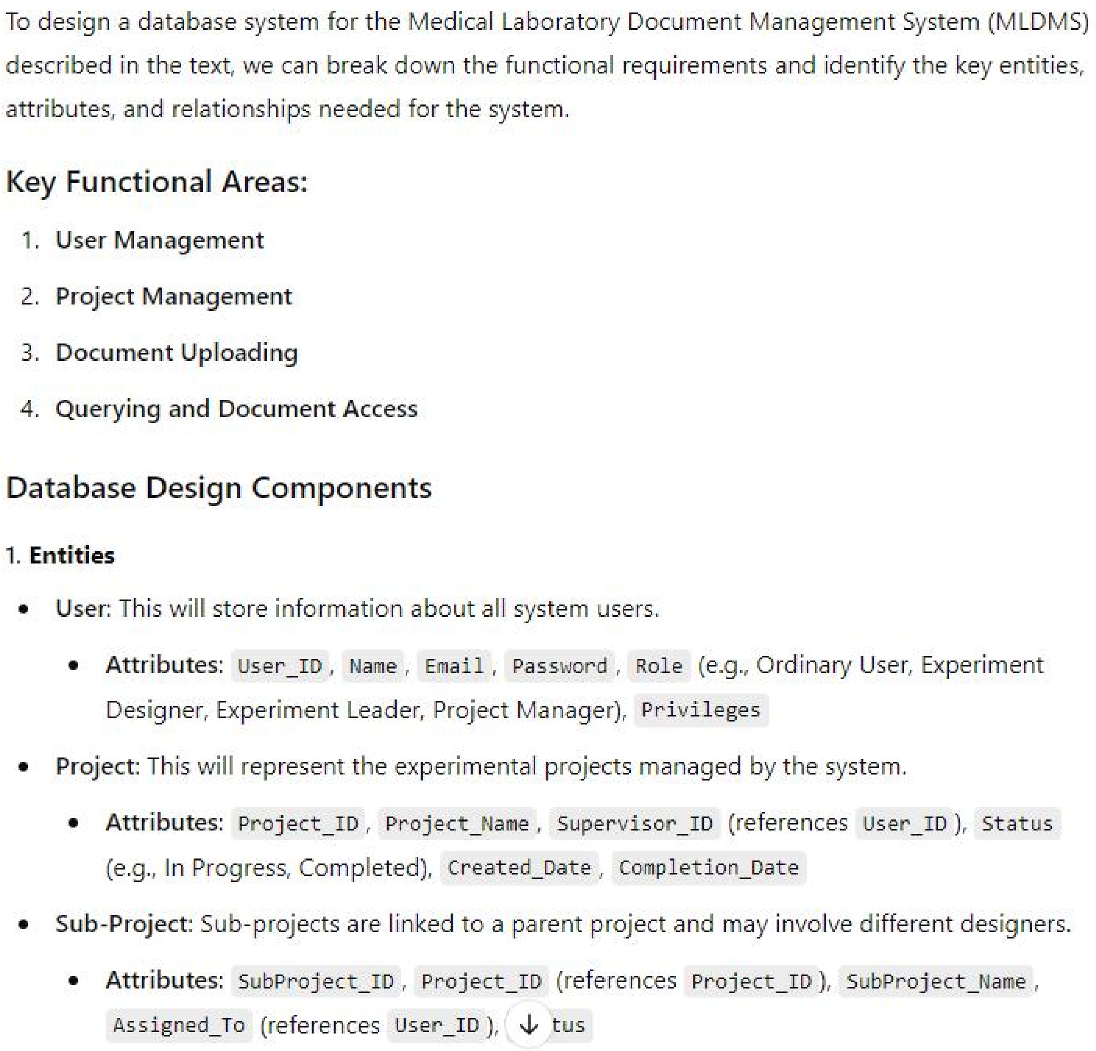
Generative AI has gradually become an important aid in the field of modern database design and data modeling. A number of academic studies have explored the potential of AI in optimizing database structures, for example, certain studies have shown that AI can automatically generate complex database models based on existing datasets. It is now safe to say that AI is not only a powerful tool that can be utilized for database design, but also a key element in driving data-driven decision making.

In modern Web development, the database, as the core of the back-end, is responsible for storing and managing a large amount of data, which directly affects the performance and stability of the application. The application of generative AI in database design and modeling can optimize the back-end structure, improve query efficiency and data processing capability, and ensure system scalability and responsiveness.

Section 2: Main Body

1. Rationalization of the structure of the assisted generation database

In designing a database for a medical laboratory document management system, I used generative AI to create a reasonable database structure. AI automatically generated key entities such as users, projects, sub-projects, documents and their relationships based on functional requirements. The AI ensured data integrity and security through the design of foreign key constraints and access logs. This approach improves development efficiency, and as some studies have noted, AI accelerates architectural design and reduces human error. However, AI-generated architectures sometimes lack a deep understanding of domain-specific requirements, and developers still need to tweak them to optimize performance. For example, AI may create problems by failing to recognize specific specialized needs when processing data, for example leading to the generation of structures that are not secure enough to handle sensitive data.



To design a database system for the Medical Laboratory Document Management System (MLDMS) described in the text, we can break down the functional requirements and identify the key entities, attributes, and relationships needed for the system.

Key Functional Areas:

1. User Management
2. Project Management
3. Document Uploading
4. Querying and Document Access

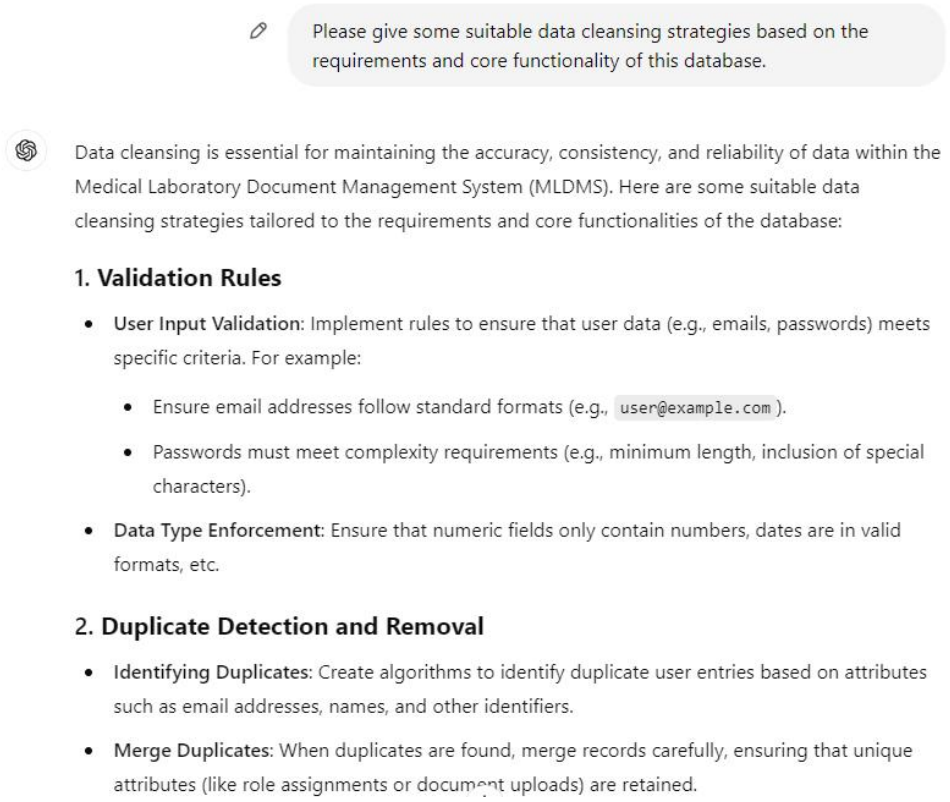
Database Design Components

1. **Entities**

- **User**: This will store information about all system users.
  - **Attributes:** `User_ID`, `Name`, `Email`, `Password`, `Role` (e.g., Ordinary User, Experiment Designer, Experiment Leader, Project Manager), `Privileges`
- **Project:** This will represent the experimental projects managed by the system.
  - **Attributes:** `Project_ID`, `Project_Name`, `Supervisor_ID` (references `User_ID`), `Status` (e.g., In Progress, Completed), `Created_Date`, `Completion_Date`
- **Sub-Project:** Sub-projects are linked to a parent project and may involve different designers.
  - **Attributes:** `SubProject_ID`, `Project_ID` (references `Project_ID`), `SubProject_Name`, `Assigned_To` (references `User_ID`), `tus`

2. Assist in data cleansing strategies

Generative AI (e.g., ChatGPT) can assist the human data cleansing process well in several ways. First, it can automate data validation by generating code snippets to enforce user-input validation rules.
Second, ChatGPT can provide algorithms for detecting and merging duplicate entries based on specific attributes, thus helping to identify duplicate data and improve data integrity. It can also facilitate standardization by proposing consistent naming conventions and formatting rules for various data types.
Third, generative AI can also analyze anomalies in numerical data by generating code, thereby supporting outlier detection through statistical methods.
Finally, generative AI can enhance user training by developing data entry documentation and best practices to empower users to maintain high data quality.



Please give some suitable data cleansing strategies based on the requirements and core functionality of this database.

Data cleansing is essential for maintaining the accuracy, consistency, and reliability of data within the Medical Laboratory Document Management System (MLDMS). Here are some suitable data cleansing strategies tailored to the requirements and core functionalities of the database:

1. Validation Rules

- **User Input Validation:** Implement rules to ensure that user data (e.g., emails, passwords) meets specific criteria. For example:
  - Ensure email addresses follow standard formats (e.g., `user@example.com`).
  - Passwords must meet complexity requirements (e.g., minimum length, inclusion of special characters).
- **Data Type Enforcement:** Ensure that numeric fields only contain numbers, dates are in valid formats, etc.

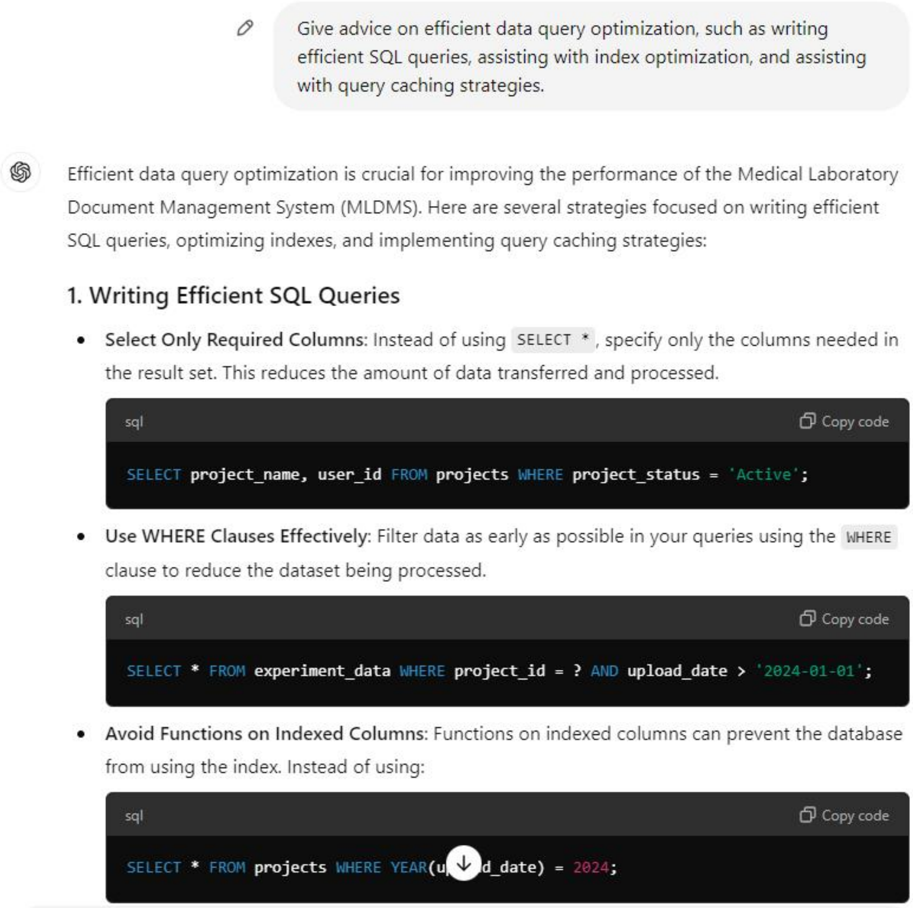
2. Duplicate Detection and Removal

- **Identifying Duplicates:** Create algorithms to identify duplicate user entries based on attributes such as email addresses, names, and other identifiers.
- **Merge Duplicates:** When duplicates are found, merge records carefully, ensuring that unique attributes (like role assignments or document uploads) are retained.

3. Give recommendations for data query optimization

Effective data query optimization is critical to improving the performance of medical laboratory document management systems because medical data is huge. Generative AI can help create efficient SQL queries, such as selecting specific columns and using appropriate WHERE clauses, to minimize data load and reduce retrieval time. For example, recommendations to avoid using functions on indexed columns can enable databases to make efficient use of indexes and optimize query execution.

However, relying on AI-generated strategies may ignore the nuances of a particular use case, which may lead to sub-optimal indexing or caching decisions. For example, while query caching can greatly speed up repetitive queries, improper cache invalidation strategies may result in rendering outdated or incorrect data. Thus, while generative AI can help identify best practices, ongoing human oversight is needed to optimize for certain unique requirements and balance efficiency with data integrity.



Give advice on efficient data query optimization, such as writing efficient SQL queries, assisting with index optimization, and assisting with query caching strategies.

Efficient data query optimization is crucial for improving the performance of the Medical Laboratory Document Management System (MLDMS). Here are several strategies focused on writing efficient SQL queries, optimizing indexes, and implementing query caching strategies:

1. Writing Efficient SQL Queries

- **Select Only Required Columns**: Instead of using `SELECT *`, specify only the columns needed in the result set. This reduces the amount of data transferred and processed.

```sql
SELECT project_name, user_id FROM projects WHERE project_status = 'Active';
```

- **Use WHERE Clauses Effectively**: Filter data as early as possible in your queries using the `WHERE` clause to reduce the dataset being processed.

```sql
SELECT * FROM experiment_data WHERE project_id = ? AND upload_date > '2024-01-01';
```

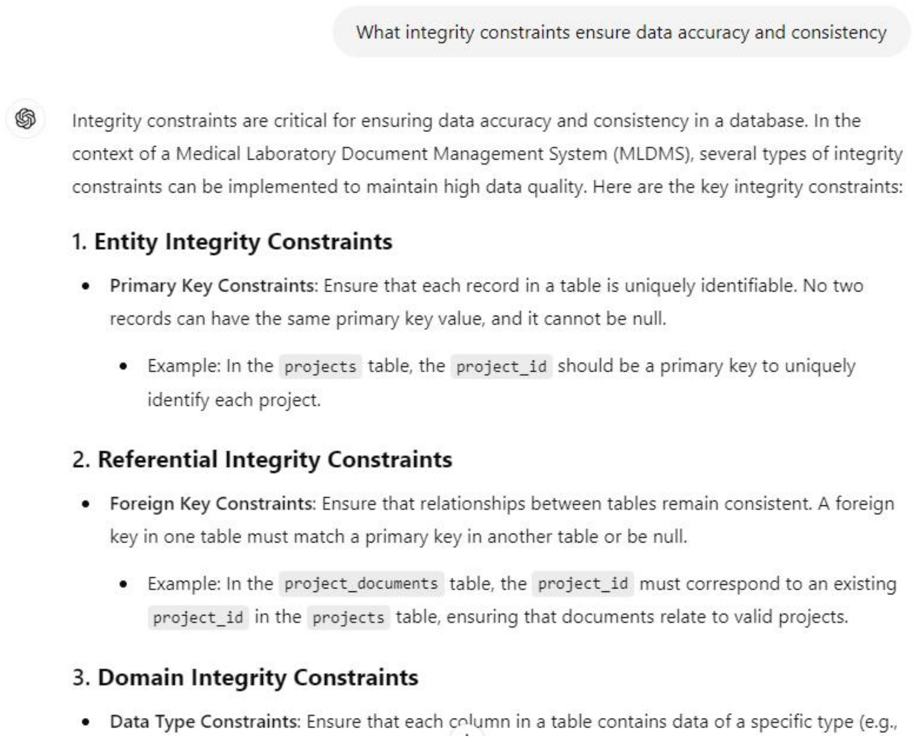
- **Avoid Functions on Indexed Columns**: Functions on indexed columns can prevent the database from using the index. Instead of using:

```sql
SELECT * FROM projects WHERE YEAR(u↓d_date) = 2024;
```

4. AI-assisted data modeling

In any data management system, the implementation of integrity constraints is key to ensuring data accuracy and consistency. Therefore I seek generative AI help to normalize integrity constraints to prevent duplicate data and inconsistent relationships to maintain data integrity. For example, setting project_id as the primary key ensures that each project is unique, and foreign key constraints ensure that documents are associated with valid projects.

However, these constraints can also present problems. Overly strict constraints can make data entry difficult and affect the user experience. In addition, maintaining complex constraint relationships may increase the administrative burden of the system, especially in the case of high-frequency data changes. The ability of AI to control this balance still needs to be improved.



What integrity constraints ensure data accuracy and consistency

Integrity constraints are critical for ensuring data accuracy and consistency in a database. In the context of a Medical Laboratory Document Management System (MLDMS), several types of integrity constraints can be implemented to maintain high data quality. Here are the key integrity constraints:

1. **Entity Integrity Constraints**

- **Primary Key Constraints**: Ensure that each record in a table is uniquely identifiable. No two records can have the same primary key value, and it cannot be null.
  - Example: In the `projects` table, the `project_id` should be a primary key to uniquely identify each project.

2. **Referential Integrity Constraints**

- **Foreign Key Constraints**: Ensure that relationships between tables remain consistent. A foreign key in one table must match a primary key in another table or be null.
  - Example: In the `project_documents` table, the `project_id` must correspond to an existing `project_id` in the `projects` table, ensuring that documents relate to valid projects.

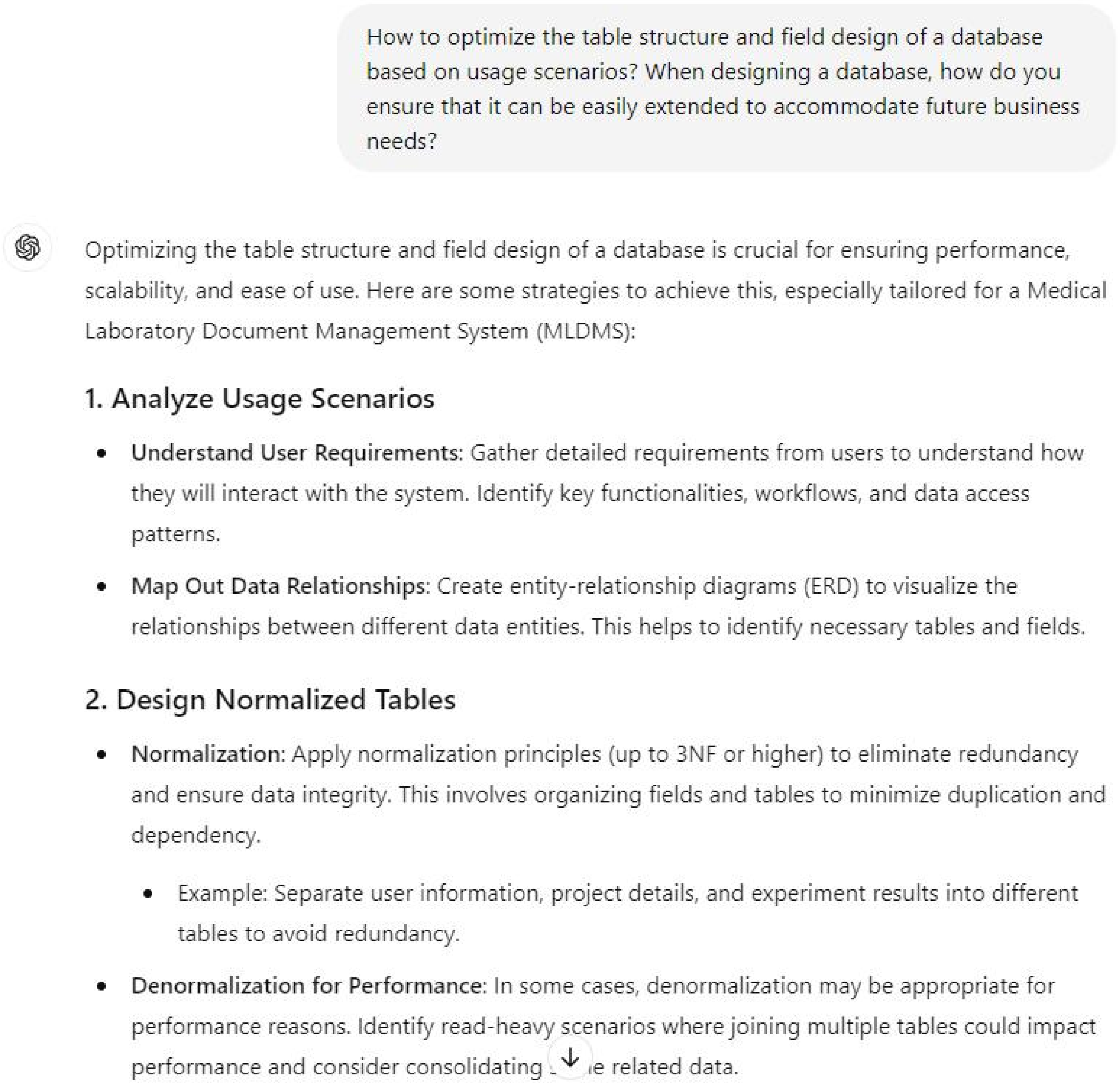
3. **Domain Integrity Constraints**

- **Data Type Constraints**: Ensure that each column in a table contains data of a specific type (e.g.,

5. Provide professional advice to meet the practical needs

Generative AI like ChatGPT can provide specialized recommendations for optimizing database design and adapting to changing business needs. For example, by analyzing usage patterns and user needs over time, the AI can make some good recommendations. For example, it can recommend appropriate ways to design table structures, indexing strategies or data types.

However, it is clearly not wise to rely too much on AI. Artificial intelligence lacks an understanding of what is really going on for a particular need or the details that a human expert might take into account. In addition, AI-generated recommendations tend to follow rigid patterns that make it difficult to get the best solution. Therefore, while AI can be an important aid in the database design process, its recommendations should still be rigorously evaluated manually to ensure that they match real-world applications and needs.



How to optimize the table structure and field design of a database based on usage scenarios? When designing a database, how do you ensure that it can be easily extended to accommodate future business needs?

Optimizing the table structure and field design of a database is crucial for ensuring performance, scalability, and ease of use. Here are some strategies to achieve this, especially tailored for a Medical Laboratory Document Management System (MLDMS):

1. Analyze Usage Scenarios

- **Understand User Requirements**: Gather detailed requirements from users to understand how they will interact with the system. Identify key functionalities, workflows, and data access patterns.
- **Map Out Data Relationships**: Create entity-relationship diagrams (ERD) to visualize the relationships between different data entities. This helps to identify necessary tables and fields.

2. Design Normalized Tables

- **Normalization**: Apply normalization principles (up to 3NF or higher) to eliminate redundancy and ensure data integrity. This involves organizing fields and tables to minimize duplication and dependency.
  - Example: Separate user information, project details, and experiment results into different tables to avoid redundancy.
- **Denormalization for Performance**: In some cases, denormalization may be appropriate for performance reasons. Identify read-heavy scenarios where joining multiple tables could impact performance and consider consolidating some related data.

Section 3: Reflection

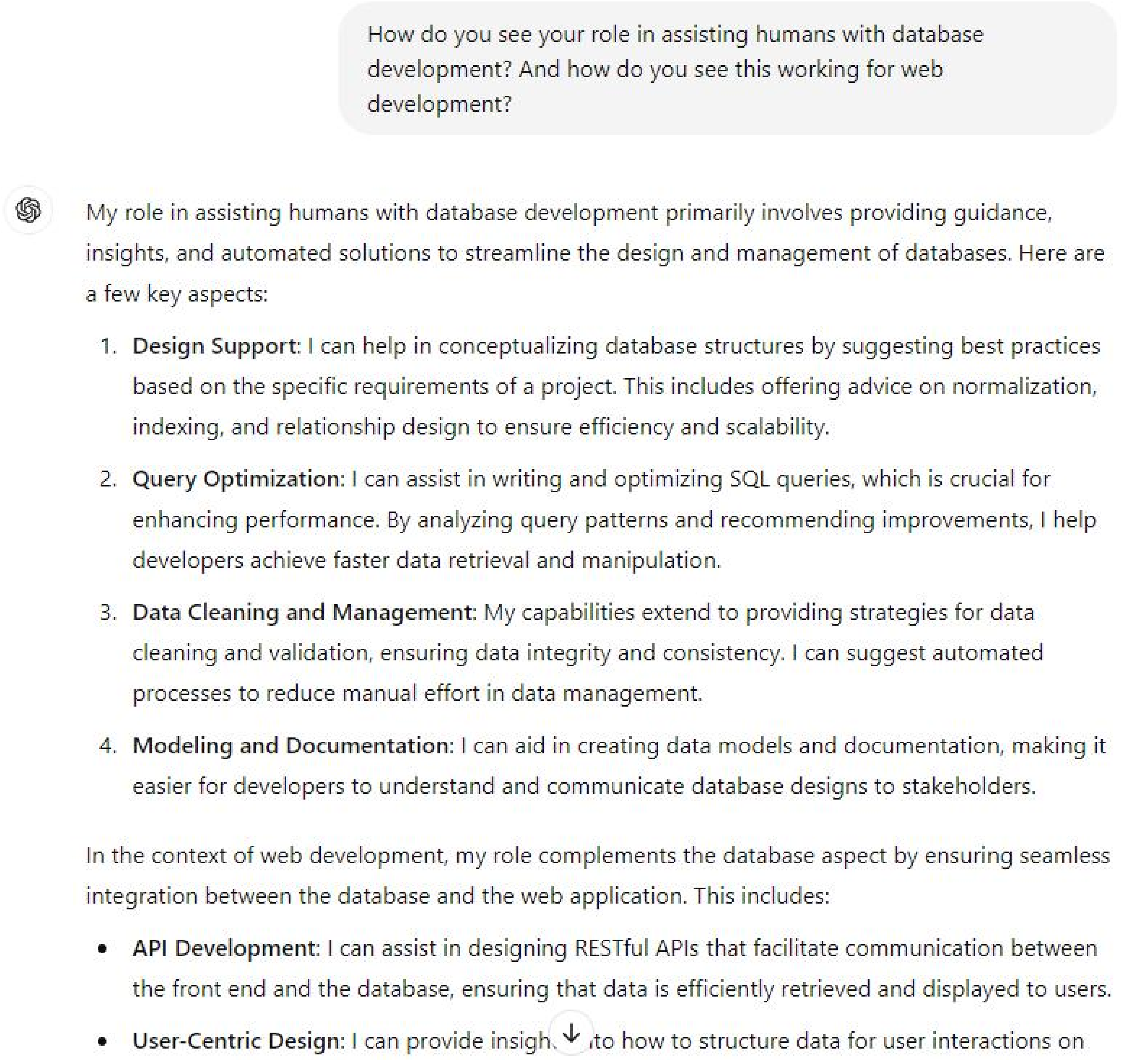
Integrating generative AI into database design greatly improves the efficiency of the back-end development part of web development. Artificial intelligence can help humans deal with business efficiently by quickly generating structured suggestions or code based on user requirements. Still, humans must be cautious with AI-generated solutions, as it follows predefined models too rigidly and it is quite possible to give a solution that is only theoretically feasible. There are also ethical concerns, especially when it comes to data privacy and security. Emphasizing data privacy in particular to AI may work for a short while, but it still requires continuous human supervisory review to reduce the risk of abuse. As technology continues to evolve, AI's capabilities and human mastery will still be in the process of being honed, and I believe that in the future, AI-assisted or even dominated database systems can be built more efficiently, securely, and flexibly.

Section 4: Conclusion

In summary, there are many benefits to using generative AI in database design, enabling developers to simplify their work. However, AI is not really intelligent and reflects the necessity of humans. The potential of this field should be further explored in the future to save developers from simple repetitive and tedious development work.

Section 5: Acknowledgement of AI use

Experiments were conducted using ChatGPT and screenshots are attached to the front. Also used to assist in figuring out ideas. The screenshots are below.



How do you see your role in assisting humans with database development? And how do you see this working for web development?

My role in assisting humans with database development primarily involves providing guidance, insights, and automated solutions to streamline the design and management of databases. Here are a few key aspects:

1. **Design Support**: I can help in conceptualizing database structures by suggesting best practices based on the specific requirements of a project. This includes offering advice on normalization, indexing, and relationship design to ensure efficiency and scalability.
2. **Query Optimization**: I can assist in writing and optimizing SQL queries, which is crucial for enhancing performance. By analyzing query patterns and recommending improvements, I help developers achieve faster data retrieval and manipulation.
3. **Data Cleaning and Management**: My capabilities extend to providing strategies for data cleaning and validation, ensuring data integrity and consistency. I can suggest automated processes to reduce manual effort in data management.
4. **Modeling and Documentation**: I can aid in creating data models and documentation, making it easier for developers to understand and communicate database designs to stakeholders.

In the context of web development, my role complements the database aspect by ensuring seamless integration between the database and the web application. This includes:

- **API Development**: I can assist in designing RESTful APIs that facilitate communication between the front end and the database, ensuring that data is efficiently retrieved and displayed to users.
- **User-Centric Design**: I can provide insigh↓ to how to structure data for user interactions on

References

1. Fui-Hoon Nah, F., Zheng, R., Cai, J., Siau, K., & Chen, L. (2023). Generative AI and ChatGPT: Applications, challenges, and AI-human collaboration. Journal of Information Technology Case and Application Research, 25(3), 277–304. https://doi.org/10.1080/15228053.2023.2233814

2. Zhang P, Kamel Boulos MN. Generative AI in Medicine and Healthcare: Promises, Opportunities and Challenges. Future Internet. 2023; 15(9):286. https://doi.org/10.3390/fi15090286

3. Pan Singh Dhoni . Enhancing Data Quality through Generative AI: An Empirical Study with Data. TechRxiv. November 07, 2023. DOI: 10.36227/techrxiv.24470032.v1

4. S. Varma, S. Shivam, B. Ray and S. Biswas, "Reimagining Enterprise Data Management using Generative Artificial Intelligence," 2024 11th IEEE Swiss Conference on Data Science (SDS), Zurich, Switzerland, 2024, pp. 107-114, doi: 10.1109/SDS60720.2024.00023.

5. Singh, J. P. (2023). The Impacts and Challenges of Generative Artificial Intelligence in Medical Education, Clinical Diagnostics, Administrative Efficiency, and Data Generation. International Journal of Applied Health Care Analytics, 8(5), 37-46.