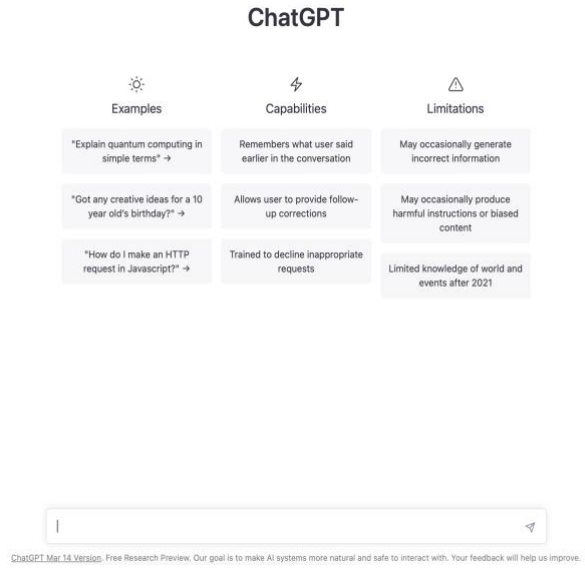
MIST 4610: Group Project 1 Information

The document below lays out information about the deliverables and the manner in which the deliverables have to be presented for the first MIST 4610 group project. I am not assigning you to a group, so your job is to get to know your classmates and form a group of 3 to 5 students.

In the first group project the team has to:

a. construct a data model,
b. build a database corresponding to that data model,
c. populate the database with data and
d. formulate 10 SQL queries (6 complex and 4 simple). These queries must be relevant from a managerial perspective (i.e., why would a manager care about these queries).
e. Every team member needs to construct/maintain a GitHub repository with details of the team project described below.

To develop the scenario for the group project the team with be working with a client, ChatGPT. You are to use the chat.openai.com interface.



You may need to create a free account or login to the interface to use it. In that chat box presented you can utilize one of the following 3 prompts:

a. Pretend you are the owner/operator of an emergency healthcare clinic needing to build a relational database. You hired some students from the MIST 4610 class at the University of Georgia to create the database for you. They need to know more about your organization to identify which entities, attributes, and relationships are important for you. Start by describing your business as a real client.

b. Pretend you are the owner/operator of a micro-loan provider needing to build a relational database. You hired some students from the MIST 4610 class at the University of Georgia to create the database for you. They need to know more about your organization to identify which entities, attributes, and relationships are important for you. Start by describing your business as a real client.

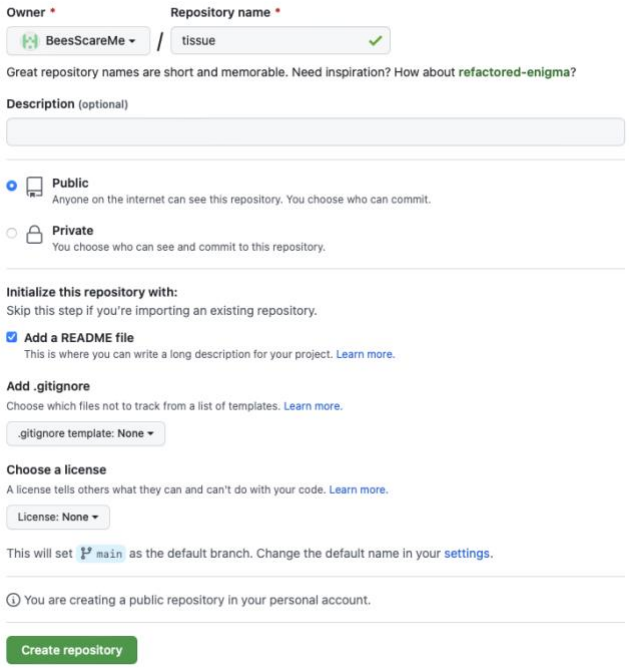
c. Pretend you are the owner/operator of a tennis (or football, soccer - your choice) club needing to build a relational database. You hired some students from the MIST 4610 class at the University of Georgia to create the database for you. They need to know more about your organization to identify which entities, attributes, and relationships are important for you. Start by describing your business as a real client.

ChatGPT should then describe based on its own large language model its understanding of the business you have specified. Based on the information and feedback provided by ChatGPT you will then take that information to create a data model with entities, attributes and relationships. You may pose further questions to ChatGPT to refine and expand your data model for instance asking it about entities, possible attributes and relationships. Ensure that the data model and business you get information on has at least 12-15 entities with an appropriate number of attributes in the entities and relationships among them. ***Please remember that the chat interaction that you have with ChatGPT should not be deleted or you will lose the history of the conversation and your conversation has to start up again.***

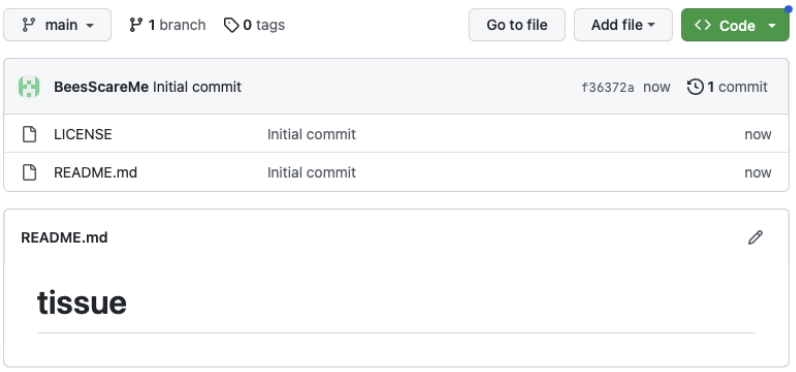
Using GitHub:

Every team member is to create a GitHub account. Creating a GitHub account is free and can be made on https://www.github.com . Sign up with an email account. Please note that you will likely need to use this account to sign back into GitHub. You can register with a free account or

a student account that may have additional benefits (you may have to verify student status). The first step will be creating a repository (or repo). This will be a public repository. You will also check the box for adding a README file. You may choose to select a license or leave the rest of the defaults in place. Please refer to image below.



After clicking on Create repository, your first GitHub repo will have been created.



In this repo you may upload project files like images and so forth using the Add file button.

The group project will be turned in using README file in GitHub to turn in documentation on an SQL-based database project. The file extension ".md" to indicate that it's a Markdown file. This

file name is case-sensitive, so make sure to use all capital letters for "README" and lowercase letters for ".md".

Why a README file?

The README file is an essential part of any project. It provides critical information about your project, such as what it is, how it works, and how to use it. A well-written README file can help other developers understand your project and contribute to it. It can also serve as documentation for end-users who want to learn how to use your database.

Submission Requirements:

Team name and members:

Include information about the name of the team, the names of all team members as well as ***links to their corresponding GitHub repos*** that have the project on them.

Problem description:

A description of the client scenario that you are modeling describing it in sufficient detail that makes sense in the context of your data model. This should be reflective of your conversation with your client.

Data Model:

Include an image of the data model (PNG file). An explanation of the data model including the relationships between the entities in natural English. Please include any screenshots of any of the substantive conversations you have had with your client that have informed the data model.

Data Dictionary:

The data dictionary explains the different columns in the different tables including data types and their roles. Please refer to sample present on ELC.

Ten Queries:

Include a natural language description of the query and a justification as to why each query is relevant from a managerial perspective (why would a manager be interested in the query results). Avoid having queries that are almost identical to one another. Add the written of the query as well as the response (e.g., copy and paste Execute to Text) of each query. You can also use the code markdown to highlight the SQL code and copy and paste the results into the file.

To ensure the complexity of the queries some of the things that may be considered include multiple table join, traditional subquery, correlated subquery, GROUP BY, GROUP BY with HAVING, multi condition WHERE, Built-in functions (e.g., AVG) or a calculated field, REGEXP, NOT EXISTS, and more.

You may combine some of the preceding list of features into a single query (still have to provide 10 queries). Indicate in matrix format in your report which features are covered in a query. A sample matrix is shown.

Database information:

	Query 1	Query 2	…….
multiple table join	x		
subquery	x	x	
…		x	

The name of the database on the MySQL server. Each table should be populated with enough data so that the queries return a sufficient result set. All queries should be bookmarked through the use of stored procedures (take a look at the example on ELC) according to this format: TP_Qx (where x is to be replaced by the query number).

Tips:

You may use the website https://readme.so to help you with the creation of the README.md file. The site is a template creator and has various sections that you may combine to form the final README.

GitHub also provides some tutorials about creating README's.

https://docs.github.com/en/repositories/managing-your-repositorys-settings-and-features/customizing-your-repository/about-readmes

See an example of a project from the previous semester:
https://github.com/taralbpatel/Mist4610GroupProject1

Grading

The final version of your Group Project will be graded. It is worth 15% of your final grade. Your individual grade will be calculated based on the team score and your contribution to the group's work. Maximum points that may be awarded for this project: 100.

Points will be distributed as follows:

The data model (30 points)
- Entities (7.5 points)
- Attributes (7.5 points)
- Relationships (7.5 points)
- Identifiers (7.5 points)

The data dictionary (10 points)
- Description of each column of each of your tables (5 points)
- Data type and length of each column in each of your tables (5 points)

The description and justification of your queries (10 points)
- Description (5 points)
- Justification (5 points)

The SQL queries (30 points or 3 points for each of your 10 queries)

The implementation of your database (15 points)
- Tables (populated), columns, keys, relationships, data (12 points)
- SQL queries bookmarked (3 points) (Remember that tables must be consistent with your data dictionary.)

The clarity of your final deliverable (5 points)

Make sure that you have included everything requested, and that your README.md is well written and well organized.

Have fun!