COMP 3380 Project Overview – Winter 2024

This project is an opportunity for you to design and implement your own database based on publicly available data. You will model the database, apply techniques, and use SQL to formulate and run queries based on concepts learned in class.

Project Teams

The project is to be completed in **teams of three people**. You may select your own team, or request your instructor place you on a team. **By January 15:**

- If you have a team of three, one of you should email your instructor with all three names and userids. Copy your teammates on the email. Your instructor will reply with your team number.
- If you do not have a team, email your instructor asking to be placed on a team. Your instructor will assign you to a team and email all three team members to connect you.

Anyone who has not emailed by January 15 will be assigned a team by the course instructors.

A note about teams: some teams will have communication problems, or work distribution problems. ALL team members are responsible for effective team functioning. In extreme cases only contact your instructor to mediate a group meeting. Waiting until just before a deadline to solve the problem is too late.

Project Deliverables

The project has several check-ins to ensure progress is being made throughout the term, and then has a final submission consisting of your project (database & interface) and a report.

Parts of the project build on each other. When starting the project, read through every part's description to have an idea of what's coming. In the final report, you will have to explain decisions you made throughout, so taking good notes throughout the term will help when writing the final report.

- Stage 1 (3%): Dataset Selection & Group Timeline – **January 26**
- Stage 2 (0%): Optional ER Diagram Check-In – February 2
- Stage 3 (8%): Database Design (ER/EER model & relational model) – **February 16**
- Reflection #1 (3%): Individual and Group Reflections, Updated Group Timeline – **February 16**
- Stage 4 (0%): Optional Query Check-In – March 1
- Stage 5 (5%): Query Design – **March 8**
- Stage 6 (5%): Interface Design – **March 15**
- Reflection #2 (3%): Individual and Group Reflection, Updated Group Timeline – **March 15**
- Project Demonstration (10%) – **April 3-10**
- Final Project Submission (35%) – **April 10**
- Final Report (25%) – **April 10**
- Reflection #3 (3%): Individual and Group Reflection – **April 10**

To help you ensure that your project is on the right track, and that you are working on something reasonable, there are two optional check-ins. We will give written feedback you can use to improve your work before you submit the next stage of the project. These check-ins will not receive a grade, but what you submit must still be of good quality so that our feedback is useful (i.e., if it's low quality, our feedback will likely be on things you already know how to improve!).

Project Marking

Please see the marking criteria in the individual part descriptions to help prioritize your work. In general, we will be looking for ambitious and creative projects that are well executed. **Details listed in the descriptions are but minimum requirements. An excellent project deserving of an A+ will seek to surpass what is described in this document**.

All grading will use a non-linear scale as follows to represent the quality of the work submitted. Each item will be graded /5. Some items might carry more weight than others.

- 5 – top of the class, amazing
- 4 – great work, all required components covered and done well
- 3 – OK work, some components missing, some things done well
- 2 – poor work, some components missing, few components done well
- 1 – very poor work, many components missing, components not done well
- 0 – not submitted or does not demonstrate understanding the deliverable

Part A: Designing a Database

Stage 1) Data Discovery (3%)

The first part of the project involves finding some data to analyze. Some viable sources are listed in UM Learn, though you are free to use other sources (e.g., wikis). Data from any public source is acceptable but remember to acknowledge your source(s)! (Be aware of any copyright or licensing issues if you want to use this data beyond this course.)

You can aggregate data from multiple sources, but all data must ultimately be connected. In other words, the ER diagram you draw with this data must be a connected graph (even better, try to have a tightly connected graph, i.e., the graph remains connected if you start removing tables). Aim to find and/or create a dataset that will ultimately break down to more than 10 tables and 1000 rows with relatively few support tables compared to main tables after completing Stage 3. A support table is a table that is usually small in both arity and cardinality, and is mostly used for lookup purposes (e.g., a table that just has Rank and Salary, where you can lookup a Salary based on Rank).

When selecting data, consider completeness of the data. A source with many blank entries will be problematic later. While generally more data (rows) will allow more interesting queries later, be careful that your dataset is not too large to easily work with.

For Stage 1, submit one pdf (one submission per group) containing:

- A 3-5 paragraph summary of your chosen dataset(s). Include information on what data is in each file (list of attributes, number of records). Indicate whether the data is ready to insert in a database or cleaning will be necessary. If cleaning is necessary, what process will you follow?
- If you plan on using more than one raw file, 1 paragraph explaining how the datasets are connected to each other.
- Links to your chosen dataset(s).
- A project timeline, outlining the tasks that each group member will complete and the deadlines for task completion. You will decide as a group how to divide the tasks among the team, and at this point may want to make a detailed plan for stages 2 & 3, and a tentative plan for the rest of the project.

Stage 1 will be graded on:

- Quality of dataset (e.g., size, connectedness)
- Completeness of data summary
- Plan for cleaning data
- Quality of timeline

Stage 2) ER Diagram (optional submission)

Look ahead to Stage 3 to see what is required in your final database design submission. The stage 2 submission is an opportunity for early feedback on your design. Feedback will focus on whether the chosen dataset is appropriate and whether the proposed ER/EER model is reasonable for later steps of the project.

For Stage 2, submit one pdf (one submission per group) containing:

- A 1 paragraph reminder of your chosen data, updated based on Stage 1 feedback, if applicable.
- An ER diagram representing your database design. The more detail included, the better the feedback will be.
- Point-form notes justifying relationships, and participation and cardinality constraints, referring to the dataset(s) as appropriate.

Stage 3) Database Design (8%)

You will draw an ER model (including EER components, if appropriate) which represents the database you have chosen to create. The model must include participation and cardinality constraints, as well as a brief justification for each constraint. Justifications should explain the “why” of constraints, not merely putting them into words (e.g., “not every Song is written by an Artist” = bad, “some Songs are written by unknown Artists, and so aren’t in the Wrote table” = better).

You will then convert your ER model to a relational model and normalize it as much as possible using the rules and standards discussed in class and in the lectures.

For Stage 3, submit one pdf (one submission per group) containing:

- A 3-5 paragraph summary of your data (e.g., a short description of what it is, how much data), updated from Stage 1 as appropriate.
- An ER/EER diagram, including participation and cardinality constraints, with text justifying your design choices.
- The final relational model (post-merge and post-normalization). Include a description of the steps you took to translate your ER model to your final relational model (i.e. steps for merging and normalizing).

Stage 3 will be graded on:

- Quality of dataset (e.g., size, connectedness)
- Ratio of support tables to main tables
- ER/EER diagram
- Justification of participation and cardinality constraints
- Translating and merging ER/EER diagram to relational model
- Normalization

Reflection #1 (3%)

As a group, submit a 3-5 paragraph reflection on your progress. Possible questions to answer might be:

- How well is the group communicating? How can communication be improved?
- Has the work been divided evenly among team members?
- Is everyone completing their assigned work on time? If not, what adjustments will be made moving forward to make the project successful?

Individually, submit a 3-5 paragraph reflection. Possible questions to answer might be:

- Did you complete the tasks you were assigned? Were those tasks completed on time?
- Will you change anything in the way you approach the project moving forward?
- Have you asked for help from your teammates when necessary?
- Is there anything you can do to support your teammates moving forward?

As a group, submit an updated timeline. At this point, you may want to make a more detailed plan for stages 4-6, and a tentative plan for the rest of the project.

Reflection #1 will be graded on:

- Quality of group reflection
- Quality of individual reflection
- Quality of updated timeline

Part B: Query & Interface Design

Look ahead to Part C, to see the expectations for your final project.

Stage 4) Query check-in (optional submission)

Look ahead to Stage 5 to see what is required in your query design submission. The stage 4 submission is an opportunity for early feedback on your design. Feedback will focus on whether the complexity and interestingness of queries is reasonable for later steps of the project (not on the correctness of SQL/code).

For Stage 4, submit one pdf (one submission per group) containing:

- A list of queries you intend to implement. For each query, give a 1 sentence (English) explanation of the query. Optionally, also include SQL and/or code that may or may not contain variables. If the latter, make sure that the variables are easily understood and/or you include comments explaining them.

Stage 5) Query Design (5%)

For Stage 5, submit one pdf (one submission per group) containing:

- A list of queries you intend to implement. For each query, give a 1 sentence (English) explanation of the query, the SQL/code, and explain why the results would be interesting to an analyst. If a query contains variables, explain what the user will be inputting.

Stage 5 will be graded on:

- Complexity of queries (at least one query which includes GROUP BY, one that includes ORDER BY, and one that includes an aggregate function)
- Interestingness of queries (would an analyst care about these results?)
- Coverage of data (is all data accessible?)

Stage 6) Interface Design (5%)

Look ahead to Part C, to see the expectations for your final project. Note that your code must be either Java or Python, and that your interface must run on **aviary** without any installs. A command-line interface is recommended (the focus of this class is, after all, on the database) but a GUI can be implemented with instructor approval.

For Stage 6, submit one pdf (one submission per group) containing:

- A 2-4 paragraph description of your interface. Include a brief description of how it will look, but also include your plan for implementation (language, command-line interface vs. GUI, etc.).

- 2-4 diagrams showing how your interface will look. Of these diagrams, 1 diagram should show your help menu/instructions for use and 1-2 diagrams should show how query results will be presented to the user.

Stage 6 will be graded on:

- Interface design/ease of use
- Feasibility of implementation

Reflection #2 (3%)

As a group, submit a 3-5 paragraph reflection on your progress. Possible questions to answer might be:

- How well is the group communicating? How can communication be improved?
- Has the work been divided evenly among team members?
- Is everyone completing their assigned work on time? If not, what adjustments will be made moving forward to make the project successful?

Individually, submit a 3-5 paragraph reflection. Possible questions to answer might be:

- Did you complete the tasks you were assigned? Were those tasks completed on time?
- Will you change anything in the way you approach the project moving forward?
- Have you asked for help from your teammates when necessary?
- Is there anything you can do to support your teammates moving forward?

As a group, submit an updated timeline. At this point, you should make a more detailed plan for the remainder of the project.

Reflection #2 will be graded on:

- Quality of group reflection
- Quality of individual reflection
- Quality of updated timeline

Part C: The Database and its Interface

See the DemoJavaProject in UM Learn for some ideas on how to get started.

Database Creation and Population

You will implement the database you designed in Stage 3. Incorporate feedback received in your final implementation. This may require revisiting your design before moving on.

You will be given some instructions in UM Learn on how to connect to a department-hosted server. Your database must be located on the server (i.e. on **uranium.cs.umanitoba.ca**, not local).

Once your database is created, you will populate it with your data. You must use a code-based method to add records, and submit that code as part of your final submission.

Implementing an Interface

You will create a front-end interface which allows a person (say, an analyst) to access and use your database. It can be as simple (e.g., command-line interface) or as feature-rich (e.g., complete GUI) as you want. However, a command-line interface is recommended, as your focus in this project should be on the database and querying. If your group wants to implement a GUI, you must first obtain approval from your instructor.

Your application should be implemented using Java, Python or as a website (i.e. index.html that is submitted as part of your project, not hosted somewhere). Your application **must run on aviary without any installs**.

When building your application, you must consider the following requirements:

- When using the interface, the database should be relatively secure according to what was discussed in class (e.g., can't allow freely entering SQL commands to prevent SQL injection).
- Your interface should support an analyst trying to answer interesting questions they might have of the data, which are not easily answerable. You might consider taking some time to come up with interesting questions an analyst might have and allow your interface to execute the relevant queries. You should support at least one query which includes GROUP BY, one that includes ORDER BY, and one that includes an aggregate function. Note that your interface does not necessarily have to make these components explicit, at a minimum, it should simply allow someone to run those queries. Some tips:
    - GROUP BY and aggregate functions are hard for humans to do on the fly, so are an easy way to create interesting queries.
    - Queries should be relevant and potentially useful to an analyst. Complex queries that are hundreds of words long, nested four layers deep could be interesting for you to implement, but might be too convoluted for an analyst to ever reasonably run.
    - Don't worry too much about optimizing your queries, as long as they run in a reasonable amount of time. However, consider informing the user with a message of some kind if a query is currently being executed (so that they don't think your interface has crashed).
- Content from all tables should be accessible one way or another. This might mean you have some simple queries in addition to the queries above.
- The interface should provide a way to delete all data from your database, and a way to repopulate it.

Project Demonstration (10%)

Each group will sign up for a timeslot during the last week of classes (April 3-10). You will give a 5-8 minute demonstration, in which you will give a brief summary of your project, and show that your project runs on aviary, connects to a populated database on uranium, executes queries, and presents results to the user. You will show that your project is robust against invalid input and SQL injection.

The project demonstration will be graded on:

- Quality of project summary
- Demonstration that interface runs on aviary and connects to uranium
- Interface functionality & ease-of-use
- Handling of invalid input

Final Project Submission (35%)

For the final project submission, submit one .zip file (one submission per group) containing:

- Everything required to create database tables, relationships, populate the tables, and the program you wrote for interacting with your database.
- Include a readme.md file with instructions on how to create and populate the database and run your program.
- For authentication, provide the userid and password for your fully-populated database. Normally this will be the userid for one of your group members.

The final project submission will be graded on:

- Code to populate database
- Quality of instructions on how to run your project
- Project robustness against invalid input & SQL injection
- Quality of code to interact with the database (interface)
- Queries – correctness
- Queries – complexity and interestingness
- Interface functionality & ease-of use

Final Project Report (25%)

For this last part, you will write a report (3-5 pages) detailing your progress through the project. If you took good notes during development, this should be relatively quick and easy to put together. The report should at least include the following information, but consider adding anything else that you find interesting or that might help someone reading your report understand what you did.

- A cover page with the names and userids of all group members
- A 1 paragraph introduction to your project

- A summary of the data: Why was it chosen? What does it consist of (attributes)? How large is it (number of records)? Was any cleaning/pre-processing required? Don't forget to acknowledge your sources! Include an ER diagram.
- A discussion of the data model:
    - Why was it broken down into those tables?
    - Did you face any difficult choices when designing the model (e.g., tricky participation/cardinality ratio decisions)?
    - Did the data model cleanly fit into the relational database?
    - Do you regret any decisions you made in your model? Did you change the model you initially designed when it came time to implement? What changes, and why?
    - Could the data be modelled in a different way, why or why not? Given the work completed, would you choose this model?
- A discussion of the database (flavour of SQL, etc.)
- A desription of your interface, including a brief description of platform/language used, and screenshots of the interface in action.
- A list of interesting queries you can run using the interface. Explain what the queries return, you don't have to include the SQL code. Explain why these queries would be interesting to an analyst.
- Concluding remarks:
    - Does this dataset require a relational database? Would other database systems be a better choice in modelling this data? Why or why not? Would the "interesting queries" you wrote be easier or harder to re-create if you were using an alternative database? Would other database systems allow for different or more interesting queries?
    - Would this database be a good teaching tool for COMP 3380? Are there good problems for future students to solve in this database?
    - A final summary paragraph
- Appendix: A summary of each group member's contributions to the project (from Stage 1 to the end).

For the final project report, submit one pdf file (one submission per group).

The final project submission will be graded on:

- Summary of data
- Discussion of data model
- Summary of the database
- Summary of the interface
- List of interesting queries
- Other interesting discussions or summaries
- Writing quality

Reflection #3 (3%)

As a group, submit a 3-5 paragraph reflection on your progress. Possible questions to answer might be:

- How well did the group communicate?
- Was the work divided evenly among team members?
- Did everyone complete their assigned work on time?
- Were you able to implement the vision that you had for your project earlier in the term? What adjustments were made to the project scope?

Individually, submit a 3-5 paragraph reflection. Possible questions to answer might be:

- Did you complete the tasks you were assigned? Were those tasks completed on time?
- Did you provide support to your teammates? How and when?
- Did you ask for help from your teammates when necessary?
- The next time you work on a team, is there anything you will do differently?

Reflection #3 will be graded on:

- Quality of group reflection
- Quality of individual reflection