

## Expectations

- You are **not** allowed to use a programming assistance tool (e.g., GitHub Copilot) nor can you use GUI generating tools (such as Java Swing UI Builder).
- There should be multiple commits in your repository that show incremental progress. We do not want to see one giant commit with all of your code.
  - We may also spot check various commits each group member has made when grading your project.
- **Your queries should give you the data required without any extraneous information i.e., we do not want you to have your application parse through (and strip down) the returned SQL query results.**
- Every SQL query must have one or more host variables as part of the expression (e.g., :empNum in the expression WHERE empID = :empNum), so that the condition is specified at run time when the user enters the value. In other words, do not “hard code” your queries in your code. Instead, allow for user input by letting the user enter (type in) some of the values to be searched for, or use a drop-down box, or buttons, etc. to allow the user to select from a few values. Any of these will provide a better user experience, and a more realistic, flexible, and dynamic application.
- **You cannot change anything in your project between the time that it is due and your demo.**

You should review the requirements for your GUI (listed in milestone 3) and the milestone 5 description (at minimum, read through the demo checklist) **before** starting the implementation.

## Deliverables

Commit the deliverables below to the CPSC 304 provided repository and submit a link to your repository on Canvas.

1. Your completed cover page, as usual.
2. A **single** SQL script that can be used to create all the tables and data in the database. If you are using multiple scripts while developing, ensure you concatenate them and hand in only a SINGLE SQL script.
  - a. Non-trivial size: make sure that your queries have some non-trivial answers (e.g., division for only 2 products is trivial)—and the same idea with aggregation (Group By), in general, because you need to have a reasonable number of groups, and you need to have some groups that have more than one row.
  - b. This SQL script should be runnable as is (i.e., if we transferred this file to the undergrad servers, we should be able to run it as is without further tweaking).
3. A PDF file containing:
  - a. A short description of the final project, and what it accomplished.
  - b. A description of how your final schema differed from the schema you turned in.
    - i. If the final schema differed, explain why. Note that turning in a final schema that's different from what you planned is fine, we just want to know what changed and why.
  - c. A copy of the schema and screenshots that show what data is present in each relation after the SQL script from item #2 is run.
  - d. A list of all SQL queries used. For SQL query requirements, check the rubric listed on Canvas for Milestone 4.
  - e. Screenshots of the sample output of the queries using the GUI (for example, you can show what data is in your table before you run the query, and then show another screenshot after running the query, from some kind of GUI input like a button).
    - i. You need only to include screenshots for the required queries – if you implemented more than what was required, screenshots are not needed for those extra queries.
4. Lastly, include a README.txt file if there's anything you want to add that's not included in your PDF file.

**Confirm with your group that the submission has happened and that it's on time.**

## TA Support

The office hours close to a deadline tend to be very busy and you may not get a chance to speak to the TA. We encourage you to start the project early so you can seek help when needed. The TAs are **not** responsible for staying behind/offering additional hours/debugging your project remotely to help you finish your project.

You can e-mail your project TA if you have questions that are not answered already on Piazza. However, vague or pre-grading questions like “Is what we are doing correct?” will NOT be answered. Please look on the resources page and Piazza for answers to common errors such as Space Quota Exceeded, PHP Not Found/Access Forbidden Error, or JDBC tunneling issues.

## FAQ

- 1. We want to use a database/programming language combination that is not officially supported by the course. Who should we notify and what should we say?**

If you wish to use anything other than Java/CS department provided Oracle or PHP/CS department provided Oracle, **every** group member must send the course coordinator an email. Please sure to also cc. your project mentor.

The email should contain the following pieces of information:

- a. The exact combo of programming language and relational database
- b. Something stating that you understand this is completely at your own risk and that we will not be able to provide any technical support whatsoever
- c. Something stating that you understand that the teaching team reserves the right to ask you to come in during the exam period to go through your project again

The email addresses for the course coordinator and your project mentor can found on Canvas.

- 2. Can I change my code after Milestone 4 is due to polish it more for our demo?**

No. This is considered academic misconduct. The code you hand in for milestone 4 should be the code you present at your demo. In your demo, you will be explicitly asked to prove that your code has not changed since the milestone 4 deadline.

- 3. I am having technical trouble doing Tutorial 6 and/or 7, how should I ask for help?**

We have **dedicated** office hours for PHP and Java help. Please check Canvas Office Hours page for the time.

**4. Do we start coding from an empty repo? Where do we start coding once we are ready?**

You can reuse the base code provided in Tutorial 6 and 7; in fact, this is recommended as opposed to starting from scratch. You must use the provisioned repository given to you by the CPSC 304 course staff.

**5. If we are building our project based on the sample code provided, do we need to cite it?**

Yes.

In general, please cite whatever code you use that is not yours. This is a good habit to get into and will save you grief later, even if it's not explicitly required in an assignment.

In the case of code we have provided, it's a great idea that you started with it and/or took lines from it. Don't go overboard, just make sure that if someone comes back to you, you can have the moral high ground.

**6. I accidentally used my personal account to commit to the given Github repository. What should I do now?**

You want to make it easy for your TAs to see what is going on and who has done the commits. Make sure that you make it clear in the documentation that you've used your personal account and what email that is.

**7. Do restrictions from previous milestones still apply (e.g., 5 tuples minimum, must have ISA and weak entity, etc.) for the project development phase (milestone 4)?**

In general, we expect that requirements from previous milestones are upheld. We understand that sometimes denormalization makes a lot of sense, particularly in cases where you won't have a ton of tuples. We would accept small tweaks but major changes need to be approved by an instructor or your project TA **prior** to the realization of the change. When in doubt, ask!

**8. Am I allowed to change some of the queries from what I stated in my milestone 2 submission?**

You can but please ensure the new query can fulfill the requirements in the same way as the query you wish to replace.

**9. How do we set up the CS server DB for our group project?**

If you are using a department provided database (e.g., Oracle), you do not have to do any server setup. If you are using a local installation of a database, then you will have to first configure the database so that your code can connect to it when running queries. The CPSC 304 teaching team does not provide any help or troubleshooting services for local database installations. To save yourself some time, we recommend using the department provided databases; the CS department provides an Oracle a server for you to connect to.

We recommend that everyone connect to the department database using their own account (as opposed to everyone using the same account).

**10. How does the SQL initialization script that has all our DDL and Insert data statements fit into the project?**

The SQL initialization script is how you can set up your database such that you have some data to work with when testing your code. Your database also needs data so that you can demonstrate the functionality and correctness of your project during the demo (milestone 5).

This script has the added benefit of:

- Providing a standardized environment for every member to test against. Every group member can run this SQL script to set up their database in the same way. This is preferable to having everyone use the same account on the database as you may get unexpected test results if multiple people are developing/testing at the same time.
- If something happens such that you cause an unwanted and irreversible change in your database, you can easily restore it to its original state. You will also not affect your other group members.

Hint: This script can be over 500+ lines long with all the CREATE and INSERT statements so start on this early. You are not expected to have a GUI button that calls this setup script (but you can if you want to!).

Note: These INSERT statements are **not** related to the GUI INSERT statement you have to make as part of the marking rubric, you need to do both so please do not get them mixed up.

**11. Do the INSERT statements used in the SQL DDL file count towards “insert operation” query requirement?**

No.