

## Course Database Project

**Assigned:** Fri 19 SEP 2025  
**Due:** Sun 26 OCT 2025  
Fri 12 DEC 2025

### Instructions:

- The assignment is to be uploaded to Canvas by the due date, which is scheduled for 11:59pm ET that day since solutions will be distributed soon after.
- We expect that you will study with friends and often work out problem solutions together, but *you must write up your own solutions, in your own words*. **Cheating will not be tolerated.** Professors and TAs will be available to answer questions but will not do your work for you. One of our course goals is to teach you how to think on your own and solve your own problems using your resources.
- We require that all work submissions be neat and organized. You may either type your solutions in a word processor and print to an Adobe pdf or write them by hand and submit a legible scanned copy in Adobe pdf. Do write and submit your answers as if they were a professional report. **There will be point deductions if the submission is not neat** (is disordered, difficult to read, scanned upside down, etc.).
- To achieve full credit, **show INTERMEDIATE steps** leading to your answers throughout.

**Purpose:** Develop a database from idea to proof-of-concept.

**Task:** One of the most effective ways to learn database development is to develop a database from a conceptual stage to its basic implementation. This assignment consists of three elements, a two-part assignment set that will provide the student with an opportunity to create a database and install that database in PostgreSQL, and the ability to present the final project in a format that is most preferred by the student.

**Students may choose to work alone or in groups up to four (4) people. Please indicate your project team plan as part of the Task 1 exercise.**

**Overall Project:** Students are expected to define a project which satisfies all the following criteria:

- The project must be able to create, read, update and delete data from the chosen database. The executed operations should be driven from user input.
- The user interface for the project may be a simple command line argument project, a web page or a GUI-based project. The user interface provides access to the database and determines the operations that are executed on the data. If you are working on another front-end project for another course, these could be combined.
- The basic assumption is that your project will use PostgreSQL as your DBMS. If you plan to use a NoSQL database or a different relational database, you must specify which database you are using in this proposal. We will work with you to define a corresponding grading rubric for your project. The teaching team will need to approve your NoSQL database choice. Seeking that approval early will provide a greater likelihood of success in the later components of the project.
- The project must demonstrate your mastery of the chosen database model (relational, document, key value, columnar etc.)

- The final result is an application that demonstrates your ability to code a system using a database as its storage mechanism as well as your ability to document the system.

The project will consist of the following tasks:

1. A project proposal (10 points)
2. Database schema as well as database programming objects - the database should be in third (3) normal form - Create data schema, export of database (schema, data, functions, procedures, triggers) (25 points)
3. A video/small group presentation demonstrating your CRUD operations (15 points)
4. A client application that accesses the database and performs all four CRUD operations (create tuples, read tuples, update tuples, delete tuples) (40 points)
5. Final report containing the final conceptual design, logical design (ERD, relational model), user flow diagram, lessons learned, and future work (10 points)

Each task listed above must be completed for the project to be accepted and granted a grade. The project accounts for 100 points and is 20% of your final grade. A total of 5 bonus points can be accrued during the completion of the project. For most students, the project is an opportunity to build a database system for a data domain in which they are interested.

#### **Task 1: Propose a database and an application (maximum: 10 Points)**

The goal of the project is to allow you to get hands-on experience with a database that is of interest to you. This is the part of the class curriculum that you design.

Submit a project proposal with the following information:

- Your name
- A top-level description of the project. Briefly describe the data and the functionality you plan to provide for the user. Consider a list of requirements for your database. What will it track? What is important to record and detail?
- SQL vs. NoSQL storage
- Software, Apps, Languages, Libraries and hardware that will be used to develop the project. Are there any machine restrictions for the project?
- Why does this project or this data domain interest you?
- An E-R Diagram of the conceptual design for the database which you will be using for the project. This diagram should contain attributes, entities, relationships, multiplicity, and the primary keys for the entities. You can use any diagramming tool you prefer. One that is freely available is <https://erdplus.com/>. You should plan on using this diagram during your class project presentation as well as in your final project write-up (where it will be graded more strictly and worth more points).
- A brief step-by-step user interaction of your application. The description should list the steps a user would be required to perform to use your application. A flow chart is sufficient.

Please create one cohesive document that contains all the above listed deliverables.

Remember this is a writing exercise, please make every effort to submit a well written proposal. A zip file with each of the individual pieces is not appropriate.

Ideas: Here are some ideas to help you consider domains and projects that might interest you.

- Create an auctioneering company database, where we need to track bidders, lot sellers, lots up for auction, and submitted bids.

- Create an Amtrak or bus company management database that tracks buses/trains, travel schedule, ticketed passengers, and drivers.
- Create a political convention database, where we need to track registered delegates from each state, the seating locations of each delegate group in the stadium for each night of the convention, the daily state delegation representative leader for each state each day of the 5-day convention, and the voting results of each state for each vote during the convention.

**Task 1 must be submitted and approved before advancing significantly into the other parts of the project. Task 1 is due by October 26, 2025, but earlier is better in this case.**

**Task 2: Database Schema and database programming objects (maximum: 25 Points)**

Provide a self-contained dump file of your database. This file should contain all necessary DDL and DML for creating your database. Please include tuples within your database so the system can be easily evaluated. (For example, it should contain the CREATE commands for the objects within your database: tables, indexes, constraints, etc. It should also contain a dump of the data as well as the user-defined functions, procedures and triggers). The schema should contain the following:

1. SQL: There should be at least six (6) tables in your SQL schema. All tables should be normalized to third (3) normal form. There should not be any unnecessary redundancy in your tables. (1-5 Points)
2. Tables should have a primary key and, if applicable, foreign keys representing relationships in your class diagram. (1- 5 Points)
3. Project is modular and uses server-side user defined functions, procedures, triggers, events. Front end code does not contain excessive SQL code. (1-5 points)
4. Provide integrity constraints such as action to be performed for foreign keys -- ON DELETE, ON UPDATE clauses within the foreign keys in the tables. Also provide additional constraints on fields that are not part of the key. (1-5 Points)
5. Error handling as appropriate. (1-5 points)
6. BONUS POINTS: Complicated schema – user data pull requires multi-joins, or many tables (> 10) due to the complexity of the data domain. (1-5 points)
7. BONUS POINTS: Identifying fields for secondary indexes. (1-5 points) Succinctly justify your index choice within code comments.
8. BONUS POINTS: Difficult process for data extraction. (1-5 points) (Total bonus points that can be allotted for the project is 5 points.)

If you choose to use MongoDB, the Database Schema requirements are below: (25 points)

1. The schema should contain all necessary data/commands for recreating your database. It should identify the containers of the system, the embedded objects and containers that may reference other documents. You should have at least one container in your MongoDB database with an embedded object or an array field. Schema should identify known optional fields. (15 points)
2. Provide integrity constraints on data stored. Also provide additional constraints on fields that are not part of the key. (1-5 Points)
3. Database programming objects defined using the MongoDB API. (1-5 points)
4. Error handling as appropriate. (1-5 points)
5. BONUS POINTS: Identifying fields for secondary indexes. (1-5 points) Succinctly justify your index choice within comments in your code. (Total bonus points that can be allotted for the project is 5 points.)

Please provide a self-contained dump of your database. The file should be named lastname\_final\_project.sql, where lastname is your family/last name for an SQL database. This file would not be part of the final report in Adobe PDF.

**Task 3: Project Presentation (maximum points: 15 Points)**

Please provide a verbal, 5-minute description of your project on video. It should contain a brief description of the project's logical schema, its architecture, and its user functionality and utility. You can provide a visual representation, such as a PowerPoint slide that describes your system and schema, but this is not required. Each person in your project group needs to be part of the presentation (e.g., presenting at least one slide). You should provide a demonstration of the front end for at least two of the CRUD operations. It is important that you be able to describe and demonstrate your project succinctly, so please do not exceed the time limit.

To demonstrate a CRUD operation:

- Use a well-known database client such as a graphical user interface (like DBeaver or Mongo, MongoBooster) to display the data in the database. Highlight the record to be updated, deleted, or read. If creating a new record, display the table in order, highlighting that the record does not exist.
- Perform the CRUD operation using your front end.
- Show the change in the database using the same database client (like DBeaver or MongoBooster), highlighting the change to the data in the database.

Please package your video into MP4 format and upload it into the assignment Canvas area along with your final report PDF. You may also submit any other content used to create the video. Please make sure all submitted files have your name as part of the file name.

Live Presentation: On December 5 and December 9, during the same period that would be class time, we will have a subset of the groups presenting their final project presentations that do not choose to do a video. If you do a live presentation, you do not need to do a video (but it can be helpful as a dry run and to have something to show to others). There will be 10 slots available for live presentations, with the mechanism to reserve a slot opening mid-November.

**Task 4: Project Application Code (maximum points: 40 Points)**

Provide one file that contains a zip file of the application code used to build your system. Please provide a README file that contains the README section from the final report.

You must have connectivity from your front end to your database in order to receive any points for this task. The host languages supported are the latest versions are Java, Python and R. Approval of any other language must be given in feedback from the project proposal.

Please be sure to include all CRUD operations (create, read, update, delete), in particular the operations that have not been demonstrated during your project presentation. If the amount of application code is excessive, you may limit the code to the functions that interact with the database and just provide a sample of the user interaction code. However, you must get this approved by the instructor before final submission.

1. A user should be able to create new tuples in the database. More points allotted for different types of entities created. (1-6 Points)
2. A user should be able to delete data tuples from the database. More points allotted for different tables supported for the delete operations. (1-6 Points)
3. A user should be able to read data from the database. More points allotted for different tables read. (1-6 Points)

4. A user should be able to update tuples in the database. More points allotted for more tables providing the update operation. (1-6 Points)
5. Functionality provided by the project is a complete solution for the chosen data domain Completeness of operations provided to the user. (1-3 Points)
6. Modularization of code use of functions in the front-end code as well as use of SQL user defined procedures, functions, triggers, and events). (1-3 Points)
7. Error handling system (testing of arguments, user input, SQL error catch/try mechanism. (1-5 Points)
8. 3 to 5 interesting queries that can be used for analysis or visualization of the data (at the detailed level or summary data). (1-5 Points)
9. BONUS POINTS: additional front-end functionality such as website. (1-5 Points)
10. BONUS POINTS: overly complicated translations from user operations to database operations). (1-5 Points)

The file should be named lastname\_project.zip, where lastname is your last name.

**Task 5: Project Final Report (maximum points:10 Points)**

Please use your project proposal report as a starting point to create your project's final report.

Provide a single document that contains the following sections:

1. Provide a README section for creating and running the project. I need complete specifications for building your project on my computer. Specify all libraries, software, etc. needed to run the application. Specify expected installation directories. If you use a specific technology for the project, the technology's download page must be listed.
2. Provide the Technical Specifications for the project.
3. Provide the current conceptual design as an ERD or Relational Model for the project as well as the logical design for the submitted database schema (Reverse Engineer your final schema in the DBeaver environment might be a good start). If you are submitting a Mongo database, please provide at least some diagram or some description of the schema.
4. Provide the final user flow of the system. List the commands or methods the user performs to interact with the system.
5. Provide a "Lessons Learned" section that contains report sections for the following:
  - a. Technical expertise gained
  - b. Insights, time management insights, data domain insights etc.
  - c. Realized or contemplated alternative design / approaches to the project
  - d. Document any code not working in this section
  - e. Provide a "Future work" section containing:
    - 1) Planned uses of the database
    - 2) Potential areas for added functionality
    - 3) No future uses or work can be documented (if justification is provided).

The file should be named lastname\_final\_report.pdf, where lastname is your last name.

Remember, this is a writing exercise. Please take the time to write a cohesive report on your semester's project.

**Submission:** Submit the results of your 5 database development parts as a single set of uploads into Canvas. The final report file attachment should be in Adobe PDF format and should be electronic and submitted as an upload to the instructor via Canvas prior to the submission deadline.