Project Deliverable

Project deliverables

Provide a zip file containing the database schema, the application code, and the final report. When submitting provide a note with your submission that contains a link to your project video. If you presented your project in class, feel free to include any content created for the presentation in your zip file. The file should be named groupname_project.zip, where groupame is your group's name containing the last names of the group members.

The maximum number of points that can be earned by the project is 105/100 points.

1. MySQL Database Schema and database programming objects (maximum: 30 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: Group of 1: at-least 5 necessary tables in your schema Group of 2-person group: 7 necessary tables in your SQL schema, Group of 3: at least 10 necessary tables in your schema. All tables should be normalized to 3rd normal form. There should not be any unnecessary redundancy in your tables.
2. Tables should have a primary key and if applicable foreign keys representing relationships in your class diagram.
3. Project is modular and uses server side user defined functions, procedures, triggers, events. Front end code does not contain excessive SQL code.
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 (
5. Error handling
6. Use of field constraints such as NOT NULL, UNIQUE, DEFAULT etc.
7. Dump submission is complete and does not require an additional submission.

Groups of 3 must complete the complicated schema bonus point along with another listed bonus work listed below.

or a MongoDB Database Schema (maximum 30 points)

NOTE: there are grade restrictions on exam 1 and exam 2 to use a different database model other than the relational data model.

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.
2. Provide integrity constraints on data stored. Also provide additional constraints on fields that are not part of the key
3. Database programming objects defined using the Mongo DB API (
4. Error handling

2. Database Application (maximum 40 points)

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, javascript 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 staff 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.
2. A user should be able to delete data tuples from the database. More points allotted for different tables supported for the delete operations.
3. A user should be able to read data from the database. More points allotted for different tables read.
4. A user should be able to update tuples in the database. More points allotted for more tables providing the update operation.
5. Functionality provided by the project is a complete solution for the chosen data domain Completeness of operations provided to the user.
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)
7. Error handling system (testing of arguments, user input, SQL error catch/try mechanism
8. Submitted project is complete. No missing files
9. Easy to use front end applications. Prompts are easy to interpret, good feedback on input.

3. Project Final Report (maximum 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 a UML for the project .
4. Provide a logical design for the submitted database schema (Reverse Engineer your final schema in the MySQL workbench). If you are submitting a Mongo database, please provide a description of your collections.
5. Provide the final user flow of the system. List the commands or methods the user performs to interact with the system.
6. Provide a "Lessons Learned" section that contains report sections for the following:
    1. Technical expertise gained
    2. Insights, time management insights, data domain insights etc.
    3. Realized or contemplated alternative design / approaches to the project
    4. Document any code not working in this section
7. 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 final report file should be named canvas_group_name_final_report.pdf, where canvas_group_name is your group name in cancas. Remember, this is a writing exercise. Please take the time to write a cohesive report on your semester's project.

4. A Project video or a presentation (maximum 10 points)

4.a Project presentation

Please provide a verbal 5-7 minute description of your project. It should contain a brief description of the project's 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. **You should provide a demonstration of the front end for at least two of the CRUD operations. If you chose to present in class and you present on the first day only 2 demonstrable operations are required, on all other days at least 3 operations must be demonstrated. 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 MySQL workbench or MongoDB compass) 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 MySQL workbench or MongoDB compass) , highlighting the change to the data in the database.

4.b Project video

Like the demonstration the video should be at most 5-7 minutes in length and contain all of the content described for the presentation. The demonstration must follow the process described for the presentation in order to receive credit. Without verification of the operation, no points will be granted.

Bonus work for the project

A grader may choose to award bonus points for the following additional features. Groups of 3 must complete at least on of these additional pieces of functionality. The maximum amount of points the project can accrue is 105/100 points.

1. 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)
2. additional front end functionality such as website or a GUI (1-5 points)
3. Overly complicated translations from user operations to database operations (1-5 points)
4. Complicated schema – user data pull requires multi-joins, or many tables (> 10) due to the complexity of the data domain (1-5 points)
5. Visualization of the data. (1-5 points)
6. Difficult process for data extraction such as web scraping or data cleansing (1-5 points)
7. Application supports multiple user roles (1-5 points)