MIS Department

MIS 531: Project Report Guidelines

Submission Requirements: Please submit an electronic copy of your project report in the due date and time, through D2L.
*Please submit a **Word document** (with the ER diagram readable in-document + openable in Visio). Do not submit a PDF/other format.*
*Additionally: submit your ER diagram as a **Visio file**. If I cannot read your diagram, I cannot grade it.*

Document formatting:

- Use a Heading 1 (Word Style) for each chapter heading in your document
- Use a Heading 2 for sub-sections within a chapter (e.g., for each query, trigger, etc.)
- Include page numbers

Contents

- Cover Page (should include group name, members, client name)
- Table of Contents (chapters along with corresponding page numbers; ensure it matches report page numbers)
  Note: start each chapter on a new page.
- **Chapter 1**: Your requirements analysis document. Include in this chapter the types of users and their requirements, and why a database approach is needed (e.g., data management issues that users face with the current system, and benefits expected with a database). Your requirements should include sufficient depth on the data to be stored for me to re-construct the same ER diagram (e.g., think of the ER homework requirements document).
- **Chapter 2**: A revised version of your conceptual schema (ER diagram) along with a data dictionary describing the semantics of each entity class. The data dictionary is the place to clarify units (such as "salary refers to salary per month") and abbreviations used. Cardinality (beyond diagram) and other integrity constraints should also be listed in the dictionary. Strong entity classes (with the exception of subclasses) should have on average 5+ attributes each.
  During Submission: Please prepare your diagram in Visio and embed it in your report document (submit both the Visio and report documents). You can use Word section breaks to setup a landscape page layout for Chapter 2 (and other chapters that require a "wide" layout).
  Your ER diagram needs to be easy-to-read (i.e., clear image that is not pixelated) for me to grade it.
- **Chapter 3**: Your relational schema (normalized) along with a data dictionary (alphabetical, but list subclasses and multi-valued attributes right after the main class table, indented). List tables in the format done in class for ER to Relational translation (mark primary keys and foreign keys).
  If you need to denormalize for final implementation purposes, you should explain your reasons and show *both* the normalized and subsequently denormalized relations (you can use a format similar to what we did in class for normalized tables).
  In your relational data dictionary: Ensure you have 5+ examples of check or unique constraints defined (i.e., in addition to primary & foreign key constraints).
  - DDL Appendix (show after Chapter 3):
    - SQL statements to create all your tables (and define constraints).

- Triggers and Procedures (code only) related to the tables (e.g., sequence/trigger combinations for primary key generation).

- **Chapter 4**: Queries. A team of 4 should have 9+, a team of 5 should have 10+ and a team of 6 should have 12+ queries for your project. In addition to the SQL for the query, please explain in English what your query does. Also show the output (if the output exceeds 25 rows or doesn't fit easily in the page, you can truncate / crop the remaining output).
  Your queries should be reasonably complex and include illustrations (non-trivial) of using joins, aggregation (Group By, Having, aggregate functions), sub-queries and inline SELECTs. Similar looking queries (similar query logic) will count as a single query. Note: using constructs where unnecessary will not count (e.g., joining with unnecessary tables, using a WITH clause where it's not needed, etc.)
  Provide a heading for each query with the query number and heading text that briefly describes the query purpose.
  You will need to input sufficient data in the relevant tables so that your queries return meaningful results. Typically, this involves 20+ rows in (each of) the queried tables. The data does not have to be actual client data ("test data" is often preferred due to privacy concerns).
  - **Note**: while you must have scripts to populate the tables (e.g., INSERT statements), you don't need to include these as part of the report (it's imperative you have a backup should there be a database problem, so you can easily recreate your schema and the associated data; there have been past cases of server outages and those who did not keep a backup were adversely affected).
  - Please use a `monospace font`, e.g., `Courier New` for the queries. Single-spaced, 10-point font. Start each clause (e.g., SELECT, FROM, …) on a new line. Use indentation for lines within a clause for readability. For example:

```
SELECT ...
FROM ...
WHERE ...
  AND ...
;
```

  Please make sure your tables and data exist in your **Group Oracle account** so I can test your code.

- **Chapter 5**: Triggers and Procedures. Your project should implement at least three triggers or procedures of reasonable complexity performing a variety of functions (i.e., if two triggers or procedures do a similar task, they count as one). Your code should demonstrate use of loops (e.g., FOR, WHILE) and conditionals (IF, THEN, ELSE, …).
  Explain in natural language the functionality of the trigger, and why it is helpful.
  At least one procedure using a cursor should be developed (this is a good place to use a loop).
  Include your code and document it to be understandable by someone unfamiliar to your project. If you are unsure about a trigger / procedure meeting the "reasonable complexity", please ask. Note: triggers that primarily generate "sequence/IDs" are not considered reasonably complex and should NOT be included in this chapter (include them in the DDL Appendix).

  Formatting recommendation for PL/SQL code: same as for SQL code (see previous chapter).

- **Chapter 6**: User Interface. For this chapter, you will create a cloud-based web interface to display the following.
  Reports: These are pages that show the results of ALL your queries that you've written in Chapter 4 (additional queries are welcome). The user should be able to easily find and execute your queries through the UI and see the results online.
  Forms: You should include facilities to add, edit and delete rows from at least 5 of your tables (including inserts/updates involving superclasses & subclasses or weak entity classes).

The report should contain a narration for a “user walk-through” including screen-captures for **every** report and data entry form. Your report should list a URL (and any user/password combination that is needed to access the system). Please highlight any features you want users to observe.
Note: All reports should be accessible through UI links or menu items. Please don’t include a link to each report / form separately in your Project Report.

Chapter 7: implementation plan. Describe an implementation plan that will describe the steps needed to implement the full project on a real-world database management system (presuming the implementing consultant has your report and design easily available). Include approximate estimates of person-hours and costs, e.g., cloud provider, personnel, hardware, software, etc. (as applicable). Break down your estimates logically and with reasonable level of detail (e.g., instead of simply saying "EC2/cloud computing costs are XYZ ", explain how you reached that total). A layout that summarizing your estimates visually is usually easier to read. You may pick any implementation platform of your choice. Provide suitable references.

- **Appendix A**: lessons learned. Describe the lessons learnt from your project experience (including what you may have learned based on presentations of other groups).

Note: Please proof-read your report for errors. Ensure all the revisions from the earlier milestone are incorporated.

Client Letter: Each group undertaking an external (live) project must request the client to directly email the instructor a letter of evaluation discussing the client experience with your project team. The letter is due at the same time as the Project Final Report. Without a letter, we assume the client was not satisfied with your team and you will be graded accordingly.