INFO9016-1 Advanced Databases

Assignment 2

Assignment: Creating a Tutorial (1)

The objective of this assignment is to create a tutorial on a database technology of your choosing that is aimed at people with a bachelor's degree in computer science or similar. You may assume that your audience has at least had courses on relational databases and some experience working with those. Think of this tutorial as a means to introduce a person to a particular database technology and encourage them to try and comprehend the database technology through a compelling running example. The tutorial should provide a brief and comprehensive overview of the database technology, including its features, benefits (and limitations), and use cases.

- Choose a database technology that interests you based on the material we have covered (so far): Datalog (e.g., XTDB) and Spatial Databases (e.g., PostGIS or Oracle Spatial and Graph). You may propose other database technologies related to this topic, but send a motivated request forapproval as soon as possible.
  - Groups who have read a paper on NoSQL databases may also propose creating a tutorial on such databases but should, again, send a motivation and request approval as soon as possible. Another option is to focus on a particular technology from one of the topics we have covered in class, e.g., spatial queries in NoSQL databases.
- Research the database technology of your choosing. You can use online resources such as documentation, scientific articles, forums, and blogs to gather information as long as they are from trusted sources and you cite them in your tutorial.
- Your tutorial should include the following sections:
  - Introduction: a brief overview of the database technology and its importance or relevance.
  - Features: a description of the features of the database technology (data types, indexing, querying, …).
  - An explanation of the benefits (and drawbacks) of (using) the database technology. You can restrict your comparison to relational databases.
  - Describe and/or cite real-world examples of how the database technology is used in different industries and applications.
  - Conclusion: a summary of the main points covered in the tutorial.
  - The tutorial: depending on your writing style, you may embed the tutorial in the various sections or dedicate a special section to the tutorial.
- Write the tutorial: Using your outline, write the tutorial. The tutorial should be well-structured, easy to follow, and engaging. Use clear and concise language, and include screenshots, diagrams, and code snippets to illustrate your points (if necessary). Important in a **tutorial is a compelling running example**. You should find or prepare a small dataset that provides a basis for simple and complex queries and pertains to one of the (types of) use cases mentioned in your tutorial. The examples you've taken may not come from existing tutorials.
- As for the submission, ideally, you can submit a self-contained tutorial (e.g., a Jupyter notebook). You may also submit a PDF or Markdown document and the software artifacts for the tutorial (e.g., a Docker image).
  - When relying on additional software artifacts, provide a README file with the instructions. When you cannot use a Jupyter notebook for the tutorial, you may provide an additional document with the expected outcomes (e.g., screenshots).

Your tutorial should be approximately 6 to 8 pages long[1] and written clearly, concisely, and objectively. You should also cite the paper and other sources you consulted at your tutorial's end. Dedicate one section to the process; how were tasks distributed and who was responsible for what task. The tutorial should be structured as **a technical report** and include: a title section (title, names, date), an introduction, a "middle," an end (e.g., conclusions or summary), and references.

You may include a preamble for setting up the Jupyter notebook; e.g., all the libraries that need to be installed or things that need to be configured for the tutorial. The preamble(s) are not included in the page limit. You may include appendices after the references. Appendices are not included in the page limit as long as their presence makes sense and is reasonable.

[1] Jupyter Notebook pages without outputs.

Evaluation Criteria

Criterion	Weight
The tutorial should cover all the required sections and provide a comprehensive overview of the database technology.	20%
The tutorial should be aimed at people with a background in relational databases. The example that has been found, curated, or created for this tutorial is compelling (i.e., “make sense for the chosen database technology”), described, and used throughout the tutorial.	20%
The tutorial itself should cover the most important aspects of the chosen database technology. The tutorial should start with a simple example (i.e., a “Hello World” query) and provide examples of more advanced features and functionalities. Each function that is presented should be introduced and motivated. Ideally, the demonstrated features provide evidence for the information presented to the reader in the various sections. You may also provide examples of things that are difficult or impossible to do.	50%
The tutorial should be well-written and free of errors. It should be well-organized, easy to follow, and should use an appropriate citation style. You may use hyperlinks instead of references when writing tutorials, but you must avoid plagiarism. In other words, make sure you adequately cite or paraphrase sources.	10%

Each criterion is graded on a scale from 0 (non-existent) to 5 (excellent). *Half points are rarely rewarded* and will be motivated. At the lecturer’s appreciation, aspects may be rewarded a 6/5 (exceeds expectations), but *these are rarely given*. The review’s overall grade is computed as a weighted average multiplied by four 4. This grade will not be rounded and will be considered as-is for your final grade.

You are encouraged to ask questions on eCampus and in class. You are encouraged to discuss your work with peers, and you are encouraged to acknowledge your peers in your review. Make sure you correctly cite and paraphrase other sources. Plagiarism and academic misconduct are not tolerated. If you are not sure about something, **ask the lecturer**. If you need help with academic writing, Justin Zobel’s Writing for Computer Science is highly recommended.