Project Research and Development

Instructions

- This assignment is a group assignment.
- The project is 25% of the total mark for the module.
- Details of the marking scheme will be communicated later.
- The estimated workload is 30 hours in total, or six hours a week for five weeks, per project member.
- Submit your paper and video (Project Group submission) by **Friday, 5 April 2024, 17:00** to EasyChair and Canvas, respectively.
- Submit your reviews (individual submissions) by **Friday, 12 April 2024, 17:00** to EasyChair.
- There is strictly no possibility of late submission.

The University takes a strict view of plagiarism and considers it a serious form of academic dishonesty. Any student found to have engaged in such misconduct will be subjected to disciplinary action by the University. Please refer to the NUS Plagiarism Policy. The University prohibits cheating in any form during assessments, tests, quizzes or examinations. Such acts will be considered, at minimum, as "Moderate" offences, resulting in the default sanction of a 'Fail' grade for the entire course.

Each group chooses one topic from the list below and indicates its choice in the GoogleSheet indicated in Canvas, and most three groups can choose each topic on a first come, first served basis.

1. Topics

    1. **Entity-relationship Graphical Interface.** Develop an interactive drawing tool facilitating the creation of Entity-relationship diagrams and corresponding SQL DDL code generation. The tool utilises an XML format for diagram representation and supports various notations, including those from lectures, UML, and logical diagrams. It allows conversion between different notations and support for multiple target DBMS and versions, accommodating system-specific syntax and additional features. Deployment options include a service, a JavaScript applet, or standalone software (e.g. in Python), with potential Cloud or virtual machine deployment.
    2. **Large Language Model for Database Design and Programming** Illustrate and evaluate the capabilities of a large language model to answer database design and programming questions. The questions may cover all or part of the syllabus of an introductory module on the design and implementation of database applications with relational database systems and SQL for computing students: entity-relationship modelling, SQL data definition, manipulation, and query languages, stored functions and triggers, and normalisation. The research involves prompt engineering and the identification of criteria that make a question easy or challenging for a large language model. The evaluation should evolve the testing of the large language model capabilities with questions from at least one popular textbook (e.g. Database Management Systems by Ramakrishnan, Raghu, and Gehrke, Johannes.)
    3. **Fake but Realistic Data.** Design, implement, and demonstrate a tool for generating realistic random data for entity-relationship designs, considering participation constraints, join selectivity, and probability distributions. The report and presentation cover background concepts such as cardinality, participation, and selectivity in relational databases.
    4. **Check Constraints Compiler.** Design and implement a PostgreSQL compiler that translates CHECK SQL constraints into triggers and stored functions. The project includes performance evaluation alongside the implementation.
    5. **Leaderboard.** Create an online platform for SQL performance competitions, allowing users to submit queries and compete based on correctness and efficiency. The platform, implemented for PostgreSQL, may feature additional functionalities such as timeouts, multiple database instances testing, and query parsing. Deployment options include a Cloud service or virtual machine.
    6. **Cost Estimator.** Design, train, and comparatively evaluate machine learning models to estimate the cost, planning time, and execution time of SQL queries with PostgreSQL. Investigate opportunities to design a machine learning model assisting the formulation of efficient queries.
    7. **Shapley Values in ProvSQL.** Build a database application utilizing ProvSQL for managing (m-)semiring provenance and uncertainty in PostgreSQL databases. The application will include support for computing Shapley values and expected Shapley values over probabilistic data aimed at exploratory analysis of probabilistic results from machine learning models. A Linux machine or virtual machine will be necessary to run ProvSQL.
    8. **Interactive Dependency Theorem Prover.** Design and implement an interactive theorem prover for dependencies, focusing on functional, Armstrong axioms, and multivalued dependencies. The project will provide a didactic background and examples, possibly following the Coq interactive theorem prover style.
    9. **Relax and Find the Key.** Develop an online game centred around identifying candidate keys of relational schemas with functional dependencies. The game involves the random generation of problems of different difficulty levels (to be defined theoretically.) The game is deployed as an app on popular game stores. The report and presentation present the theoretical background and the game.
    10. **Monte-Carlo Sampling Normalisation.** Investigate the distribution of minimal covers and normal forms of a relation with functional dependencies using Monte Carlo sampling techniques. The project involves understanding and implementing algorithms for the uniform generation of sets of functional dependencies at random, the computation of minimal covers, and the testing of normal forms.
    11. **The Chase.** Create an interactive tool implementing the Chase algorithm for functional and multivalued dependencies, offering functionalities like entailment, lossless decomposition, and minimal cover generation. The tool, deployable as a service or standalone software, provides theoretical background coverage and may offer multiple Chase algorithm versions. The project includes examples.
    12. **Xpath for JSON.** Design, implement and demonstrate an XPath-like language for querying JSON data, potentially extending to an XQuery-like language. The project includes a library implementation

for Python or JavaScript, covering language syntax and semantics in the report and presentation. The project includes examples.

13. **Datalog Compiler.** Develop a compiler translating Datalog queries to SQL, targeting PostgreSQL. Additionally, create a Python library for defining and executing Datalog queries. The project provides educational content on Datalog in the report and presentation.

2. Paper

   Your work and results are presented in a paper. The paper's submission is managed as a scientific and technical conference. The paper consists of six (6) sections, followed by a list of references.

   (a) The *Title* summarises in a few words the main idea of the work.

   (b) The *Abstract* overviews in a few lines the motivation and work and announces the main result or results.

   (c) The *Introduction* section presents the challenge and outlines the work.

   (d) The *Background* section syntheses the domain and technical textbook knowledge necessary to understand the paper.

   (e) The *Related Work* section surveys the related work in order to justify the choice of the selected techniques.

   (f) The *Methodology* section presents the selected techniques.

   (g) The *Performance Evaluation* section presents the experimental set-up, including data sets and metrics, and presents and analyses the results of the comparative empirical performance evaluation.

   (h) The *Conclusion* section summarises the work and results.

   (i) The *References* lists the sources cited in the paper.

   The paper is no longer than fifteen (15) pages, including figures and references, in Portable Document Format (`.pdf`). The paper is written in LaTeX[1] using Overleaf [2] (`www.overleaf.com`) and follows Springer Lecture Notes in Computer Science's template. The group leader submits the paper (see Canvas to find out who your group leader is) to the CS4221/5421 conference created for this purpose in EasyChair[3] (`easychair.org`.) You will receive an invitation to `easychair.org` and to the CS4221/5421 program committee in due course.

3. Video

   The video summarises the work and results as would the presentation of a paper in a scientific and technical conference.

   (a) The video consists of slides and comments

   (b) The video follows the same structure as the report.

   (c) The video may include other relevant content such as demonstrations of the models.

   The video must be in MP4 (`.mp4`) or in QuickTime (`.mov`) file format no longer than 10 minutes and no larger than 200MB (you may use `handbrake.fr` to reduce the file size, if needed.)

4. Reviews

   The project's evaluation is managed as a scientific and technical conference.

   (a) Each student is assigned to review three papers.

   (b) Download the papers assigned to you in CS4221/5421 conference in EasyChair.

   (c) Read and review the papers following the guidelines that will be communicated to you.

   (d) Submit your reviews to the CS4221/5421 conference in EasyChair.

[1] LaTeX is a typesetting system widely used for scientific and technical writing.

[2] Overleaf is an online collaborative platform for LaTeX document creation. Overleaf has a user-friendly interface that simplifies LaTeX usage, enables real-time collaboration and facilitates access to various templates and tools.

[3] EasyChair is an online conference management system designed to streamline the submission and review process.