Project: Research and Development

Instructions

- The maximum mark for this project is 30 points for 30% of the final mark.
- This project is a group project.
- Register your team of up to five members in Luminus by **Friday 11 March 2022, 21:30**.
- Research the project topics and follow the project topics presentation and Q&A session on **Thursday 10 March 2022, at 19:00** in order to make your choice of a topic.
- Submit your preferences by email (only one email per team) indicating your team's ranking of the project topics in strict decreasing order of preference to `steph@nus.edu.sg` by **Friday 11 March 2022, 21:30**. Your team will then be assigned a project topic considering the teams' preferences.
- Download the project paper template from Luminus:
  "`Files > Projects > Research and Development` ".
  It is recommended that you create a new project in Overleaf, `overleaf.com`, by uploading the ZIP file provided, to write and edit your project paper.
- Submit an outline (two pages in PDF free format) of your project plan to Luminus:
  "`Files > Projects > Research and Development > Outline Submissions`"
  by **Friday 18 March 2022, 17:00**.
- Submit your project paper in PDF to:
  `https://easychair.org/my/conference?conf=dadt2022`
  by **Friday 8 April 2022, 17:00**.
- Submit your presentation video to Luminus:
  "`Files > Projects > Research and Development > Video Submissions`"
  by **Friday 8 April 2022, 17:00**.
- There is strictly no late submission.
- Note that you will be asked to review your colleagues' project papers, and to write and submit your reviews to EasyChair by **Friday 15 April 2022, 17:00**.

This is a list of projects that your team needs to choose from. Each project should be further defined, adapted, and focussed in discussion with the teaching team.

After a project is assigned to your team, submit an outline of your plans for the project (further definition and focus.)

1. Design and implementation of a relational algebra graphical editor and its compiler or interpreter (translate into SQL) for PostgreSQL.
2. Design and implementation of a tuple or a domain relational calculus compiler or interpreter (translate into SQL) for PostgreSQL.
3. Design and implementation of a Datalog compiler (translate into SQL) for PostgreSQL.
4. Design and implementation of a Query-by-Example graphical editor and interpreter for the interactive exploration of star schema databases with PostgreSQL.
5. Design and implementation of a tool that generates realistic random data for an entity-relationship design considering participation constraints, join selectivity, probability distributions, and joint probability distributions.
6. Design and implementation of a CHECK constraint compiler for PostgreSQL that translates CHECK constraints in SQL into triggers and stored functions.
7. Design and implementation of a ladder board Web service for the submission and automatic evaluation (according to performance or results) of SQL queries with PostgreSQL.
8. Comparative feature and performance analysis and evaluation of object relational Mapping toolkits for Python.
9. Comparative performance analysis and evaluation of PostgreSQL levels of transaction isolation.
10. Design and implementation of a compiler for an XPath dialect for JSON and MongoDB.
11. Design and implementation of an XML graphical interactive exploration tool with XPath support with eXistDB.
12. Design and implementation of an interactive theorem prover for functional and multivalued dependency.
13. Counting functional dependencies minimal covers and normal forms. You may try and prove analytical bounds or use Monte Carlo methods to compute empirical results.

Submit a report presenting your project work and results. Record a video presentation of your project work and results. The code and other ancillary documentation is expected to be available to the teaching team (e.g. on GitHub.)