# *Design and Implementation of a Prototype Data Processing System*

## INF670E – Project

Angelos-Christos Anadiotis

# Logistics

- 32 students enrolled
  - 8 teams of 4 people
- The project requires to:
  - Study the literature and existing systems
  - Design and implement data structures and algorithms
    - Single node/thread
    - Parallel/Distributed

# Tasks

1. Create a relational database which follows the TPC-H schema and keep it in the main memory
2. Store the database to the disk
3. Implement the following operators:
    1. Projection (with duplicate elimination)
    2. Selection
    3. Join
    4. Group by
    5. At least one aggregation function of your choice (e.g., SUM)

# Related work

- Which algorithms/data structures do existing DBMS use?
- Which algorithms are fundamental in the related literature?
  - Check DBMS-related venues like (non-exhaustive list)
    - Conferences: SIGMOD, VLDB, ICDE, EDBT, CIDR
    - Journals: ACM TODS, IEEE TKDE, Elsevier Information Systems, The VLDB Journal
  - Browse the web to find popular algorithms and data structures

➢ It is important to separate out the fundamental works from the enhancements

# Design and implementation

- Select the best algorithm for every operator
- Design and implement the algorithms and the data structures in two steps:
  1. Single-threaded execution
  2. Multi-threaded execution
  3. Distributed execution (Spark)
- Design and implement the storage in two steps:
  1. Following the row-store approach
  2. Following the column-store approach

# Datasets and Workloads

- Use TPC-H as the base schema for the database
  http://tpc.org/tpc_documents_current_versions/current_specifications5.asp
- Use a small scale factor in order to fit the dataset in the main memory
  - You can use a bigger one when you store data to the disk
- You can use queries from the ones given in TPC-H, but you can also come up with your own
  - Use at least 3 queries and explain why you picked them

# Deliverables

- Report including
  - The design of your systems
  - Experimental results
  - Comparison of the different approaches that you implemented
- Source code
  - Github repository of the team (to become public)
    - Carefully pick your license (e.g., MIT?)
  - Send me the files with email
- Presentation of your results
  - Each team will get its own time slot
  - All team members will be asked questions

# Project grading

- [50%] Baseline implementation:
  - 1 thread
  - Any storage layout, stored in the main memory
  - All operators
- [15%] Spark support
- [10%] Multi-threaded support
- [10%] Column-store & Row-store support
- [10%] HDFS support
- [5%] Disk support

# Course grading

- [80%] Project grade
- [20%] Oral exam grade
  - Presentation
  - Question answering: questions may span across the whole course

# Deadlines

- Project delivery (including all material except the presentation)

  Friday, October 29 2021 23:59 Paris time

- Presentation / defense

  Thursday-Friday, November 4-5 2021 / TBD

- In case everything is uploaded in Github, the last commit until the deadline will be considered
  - Otherwise, send me the files with an email until the deadline

➢You may use Github at a later stage to publish your work

# Project Registration Deadline

Register your team until

**Monday, October 11, 2021, 23:59 Paris time**

# Good vs Bad things to do

- **Good**: help each other within and across teams
  - Explain at an approach level
- **Bad**: share your source code with another team
  - Also, don't get your code from online sources like stackexchange – it is easy to figure out by comparing with the rest of the code
- **Good**: If you are not sure about some parts of the work, e.g. regarding the dataset or the workload, ask me
  - Don't wait until the very last moment, as replies will not be prompt
- **Bad**: Use existing libraries or software
  - You will be graded based on the implementation of the algorithm using the standard Java and Spark API and not any other libraries, including SparkSQL