UNIVERSITE LIBRE DE BRUXELLES September 2015
Ecole Polytechnique

INFO-H-417 : Database System Architecture Course Information

The objectives in brief

In contrast to a typical introductory course in database systems where one learns to design and query relational databases, the goal of this course is to get a fundamental insight into the implementation aspects of systems designed to manage and process large amounts of data. Our objective in this respect is two-fold. (1) To gain the background required to design and implement future data management and processing systems and (2) to gain an understanding of how performance of practical data management systems can be tweaked.

In particular, we take a look under the hood of *relational* database management systems, with a focus on *query and transaction processing*. The focus on relational database management systems is motivated by the fact that the algorithms and architectures underlying relational databases have strongly influenced the design of contemporary data processing and management systems : graph databases, in-memory database systems, stream databases, and even NoSQL systems.

With respect to query processing, we study the whole workflow of how a typical relational database management system optimizes and executes SQL queries.

With respect to transaction processing we study how a typical relational database management systems ensures recovery from errors and controls concurrent access to the data.

Contacts

— Lecturer : Stijn Vansummeren (UB4.125, `stijn.vansummeren@ulb.ac.be`)
— Assistant : Stefan Eppe (UB4.131, `stefan.eppe@ulb.ac.be`)
— Course Web Page : `http://cs.ulb.ac.be/public/teaching/infoh417`

Schedule

Please check the course online calendar available on the course website schedule and room information.

Detailed Objectives

Upon successful completion of this course, the student should master the following competences :

1. Translating a given SQL expression into the Relational Algebra
2. Improving a relational algebra expression by, where possible, removing redundant joins in select-project-join subexpressions
3. Improving a relational algebra expression by, where possible, (a) replacing cartesian products by joins ; and (b) pushing selections and projections
4. Describing and being able to implement traditional secondary-memory index structures (BTrees, Hashing)
5. Being able to describe and demonstrate the shortcomings of traditional index structures with respect to multi-dimensional search keys. In addition, explaining the studied multi-dimensional indexes by means of an example

6. Describing the most important implementation algorithms (one-pass, sorting, hashing, index) for each of the relational algebra operators, as well as judging the cost of each operator, and knowing their limitations of applicability
7. Given a logical query plan and given base statistics about the size and distributions of the database relations, constructing a heuristically optimal physical query plan, by estimating the sizes of the intermediate results and correspondingly comparing the possible implementations. When joins can be reordered, choosing the order with the least cost.
8. Solving exercises on logging like the ones in sections 17.2.6, 17.3.5 and 17.4.4
9. Solving exercises on concurrency control like exercises 18.2.4, 18.2.5, 18.8.1, 18.8.2, and 18.9.1
10. Solving exercises on recoverability like exercises 19.1.1, 19.1.2 and 19.1.3
11. Being able to reconstruct the studied proofs

Method of evaluation

The project work contributes 6/20 points to the overall score, and the written exam contributes the remaining 14/20 points. Participation in both the project work and the written exam are mandatory requirements for passing the course.