

Abstract

In this project, students work in teams to implement some of the algorithms used for query optimization.

Project proposal

The class will be cut into a few project teams, each with four students. Each team will select a small database project, come up with a design, and complete the project by the end of this semester. The proposed design and group members name must be sent by email during the first week.

The implementation is to be done using the language of choice (preferably c/c++) and a detailed report, presentation and source code must be submitted at the end of the project. The project teams will make a presentation to the whole class.

The group should go through the following steps in completing this project:

Step1:

1. Design the database, containing two relations R and S and a one-to-many relationship (e.g $R.A=S.A$).
2. Creating two data files containing the data of the relations. The team has to include the functionality to create and read from these files.

📌 **Note:** Use only simple data types for attributes like integers or arrays of characters.

Step2: Fill the files with a large amount of data.

Step3: Build a B+-tree file on the data files: one uses the primary key of R and the second uses a secondary key in S (hint: choose the attribute that is participating in the relationship).

Step4: Find the time cost for searching a key using the B+-tree files and compare it to the time cost using file scan.

Step5. Perform a natural join on the relations. **Show the resulted number of records and the time cost.**

📌 **Note:** The final documentation should include the algorithms used, the tree/data files structure, screenshots and statistics (e.g cost of the applied operations and the number of records in the resulted relation).