Access path selection in a relational database management system [5].

ACM Reference Format:
. 2020. Access path selection in a relational database management system [5].. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 ABSTRACT / INTRODUCTION

1. What opportunities/changes that make this work useful and timely? 2. Why existing approaches fail to make use of these opportunities? 3. How do you propose to do better? 4. Why this problem is relevant to the course? (1-2 sentences each)

SQL is a declarative alternative to imperative data manipulation languages used by IMS and proposed by CODASYL. SQL translates into relational algebra (*RA*), which is built on top of a theoretically sound relational data model. SQL is good because it makes the programmer's life easier by providing data independence so that changes in how the data is physically stored does not affect the SQL queries that developers write to access and manipulate their data.

This development is important because developers currently spend most of their time writing and re-writing their applications whenever there is a change to the schema of the data or how it is stored. This means developers don't have time to build new features and improve their applications. SQL can help address this issue, but current SQL execution engines are unacceptably slow and nowhere near the performance of hand-tuned IMS/CODASYL queries.

SQL can translate into many different, but semantically equivalent, relational algebra statements, and each statement can be executed by many possible *query plans*, it is desirable to pick the plan that is fastest to run. This project proposes a way to estimate the cost of executing a query plan, and automatically search the space of all valid query plans for a SQL query to find one that is fast no execute.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

This problem is relevant to the course because data visualization systems are often built on top of data management systems, so making the data management system faster would thus make the data visualization faster.

2 ONE SENTENCE SUMMARY

Describe your project in one sentence, in other words, your hypothesis.

We will devise a cost model to give each query plan a score, and use existing equivalence rules to define and search the space of alternative plans to find one with low cost. Our hypothesis is that this will be competitive with hand-coded execution plans.

3 AUDIENCE AND NEEDS

Who are the audiences for this project? How does it meet their needs? What happens if their needs remain unmet?

This project will impact the academic community as well as every application developer. The community benefits because it will validate the hypothesis that declarative lanugages such as SQL can run quickly, and open up a new field of research in data management.

Application developers benefit because they will not need to write imperative data manipulation code, worry about how to hand optimize the code to be fast, and can focus on building application features.

4 APPROACH

What is your approach? Why do you think it's a good approach and will be successful?

(1) We will build cost models for individual operators. Recent advances on disk-based indexes [2] suggest that we will need different cost models depending on how the data is stored. We may need to make simplifying assumptions and model the costs based on very simple statistics.
(2) We will combine the per-operator cost models estimate the cost for an entire query plan.
(3) With a cost model, we can now search through the space of all query plans that return the same results. This space is huge, so we will develop heuristics to reduce the space. Past experience with IMS has shown that the dominant cost will be combining multiple

relations using joins so we will focus on them. The space of possible joins can also be very large, so we will develop a dynamic programming-based approach to search through the space.
(4) Run experiments and show that the plans picked by the optimizer are reasonable.

5 (BEST CASE) IMPACT

In the best-case scenario, what would be the impact statement (ideal outcome and conclusion) for this project?

We can show that the optimizer picks reasonable plans that are comparable to hand-optimized plans selected by expert developers.

6 MILESTONES

List all major milestones for this project

(1) Identify the list of statistics that we can reasonably compute and store in tens of kilobytes of space.
(2) Develop a cost model for individual operators and show that the model is correlated with reality by running them on synthetic datasets based on what existing IMS customer use.
(3) Develop a cost model for a full query plan.
(4) Derive an estimate of the size of the full plan space for a given query plan and show that it is infeasibly large.
(5) Develop dynamic programming heuristic to search the plan space.
(6) Run experiments on synthetic datasets to compare the optimizer-picked query plans with hand-optimized query plans.

7 OBSTACLES

*What are the major and minor obstacles that could happen? Note that major obstacles are situations where you would consider **killing** the project. Minor obstacles are things that would delay the project or increase the overall cost in energy, time, people, and money.*

7.1 Major obstacles

- If we cannot show decent correlation between individual operator cost models and reality, then the rest of the project may not work. We will also need to define what "decent" means.

7.2 Minor obstacles

- We may not have access to computing resources to run any experiments, in which case we will need to focus on theoretical aspects of the work. This could reduce the impact of the project, however showing that an optimizer is *possible* is still a contribution.

8 ADDITIONAL RESOURCES

What additional resources do you need to complete this project?

- Some computational time to run our optimizer algorithm to generate some query plans.
- Access to a machine where we can install and run experiments using our current database prototype.

9 LITERATURE REVIEW

List 5 major publications that are most relevant to this project, and how they are related.

- *Background for the project:* This work builds on prior work on relational algebra and the relational model [3], and on new relational database systems [1, 6]
- *Work the project relies and builds on:* Some preliminary work has suggested a language for specifying query plans [4] that we could borrow from. Also, Recent work [2] on different ways to store and index data lend credence to the need for different cost models for access data in relations.
- *Direct competitors:* We could not find existing works on alternative techniques to automatically optimize query plans.
- *Alternatives to achieve the broader goal:* As stated above, IMS and CODASYL [7] are the main alternative data management systems, but they do not have any automated query optimization.

10 DEFINE SUCCESS

When / How do you know if you have succeeded in this project? In other words, what is the minimum finding that would make this project a success and publishable?

Simply developing a set of cost models and search heuristics for query plans should be publishable, because an automated optimizer of any sort does not yet exist.

REFERENCES

[1] M. M. Astrahan, M. W. Blasgen, D. D. Chamberlin, K. P. Eswaran, J. N. Gray, P. P. Griffiths, W. F. King, R. A. Lorie, P. R. McJones, J. W. Mehl, et al. System r: relational approach to database management. *ACM Transactions on Database Systems (TODS)*, 1(2):97–137, 1976.
[2] R. Bayer and E. McCreight. Organization and maintenance of large ordered indexes. 1972.
[3] E. F. Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970.
[4] R. A. Lorie and J. F. Nilsson. An access specification language for a relational data base system. *IBM journal of Research and Development*, 23(3):286–298, 1979.

[5] P. G. Selinger, M. M. Astrahan, D. D. Chamberlin, R. A. Lorie, and T. G. Price. Access path selection in a relational database management system. In *Proceedings of the 1979 ACM SIGMOD international conference on Management of data*, pages 23–34. ACM, 1979.

[6] M. Stonebraker, G. Held, E. Wong, and P. Kreps. The design and implementation of ingres. *ACM Transactions on Database Systems (TODS)*, 1(3):189–222, 1976.

[7] R. W. Taylor and R. L. Frank. Codasyl data-base management systems. *ACM Computing Surveys (CSUR)*, 8(1):67–103, 1976.