Technical Perspective: Query Answers – Fewer is Faster

Leonid Libkin
The University of Edinburgh & RelationalAI
Edinburgh & Paris
l@libk.in

We often write queries using LIMIT k, indicating that only k answers are to be returned. This feature is present in most query languages, for different data models: SQL, SPARQL, Cypher etc. For example, in a repository of about 250M SPARQL queries, about 15M queries are of this form. Not surprisingly of course, the database research community studied such queries extensively. The dominant setting is this: there is an ordering on tuples that can be returned by a query. Then the answer is limited to the first k tuples in this ordering.

But how realistic is the ordering assumption? And even if an ordering on tuples can be defined somehow, how important is it for query answering?

To use an ordering explicitly to select the top k answers, one would combine the LIMIT clause with an ORDER BY clause. But is this combination really the prevalent one in real-life queries? It turns that there are important cases (again, based on a detailed analysis of the above mentioned SPARQL query log) in which the answer is negative. In fact, fewer than 0.1% of queries that use LIMIT also use ORDER BY. Furthermore, in queries that use LIMIT k, the number of returned tuples is usually small: k is typically under 100.

There is therefore a tantalizing question: if we do not really care about the order in which the first k answers to a query are returned, can this be used to our advantage? In particular, can this be used to evaluate queries faster? This is precisely the question that the VLDB 2022 paper “Threshold queries in theory and in the wild” set to answer.

At first, it explained a variety of scenarios where such queries appear. These include data exploration (e.g., show me some random 10 tuples from the query answer), cardinality constraint queries (find violations of cardinality restrictions, e.g., limits on sizes of groups), or data monitoring queries (say, connecting a group to a number of related features). Having provided examples of those queries written in SQL, SPARQL, and Cypher, the paper then went ahead with a combination of theoretical and experimental studies.

Theoretical study.

The key contribution here is a clean theoretical model that encompasses the above types of queries and is at the same time amenable to complexity-theoretic and algorithmic analysis. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Copyright 2008 ACM 0001-0782/08/0X00 ...$5.00.

Technical Perspective: Query Answers – Fewer is Faster

Leonid Libkin
The University of Edinburgh & RelationalAI
Edinburgh & Paris
l@libk.in

We often write queries using LIMIT k, indicating that only k answers are to be returned. This feature is present in most query languages, for different data models: SQL, SPARQL, Cypher etc. For example, in a repository of about 250M SPARQL queries, about 15M queries are of this form. Not surprisingly of course, the database research community studied such queries extensively. The dominant setting is this: there is an ordering on tuples that can be returned by a query. Then the answer is limited to the first k tuples in this ordering.

But how realistic is the ordering assumption? And even if an ordering on tuples can be defined somehow, how important is it for query answering?

To use an ordering explicitly to select the top k answers, one would combine the LIMIT clause with an ORDER BY clause. But is this combination really the prevalent one in real-life queries? It turns that there are important cases (again, based on a detailed analysis of the above mentioned SPARQL query log) in which the answer is negative. In fact, fewer than 0.1% of queries that use LIMIT also use ORDER BY. Furthermore, in queries that use LIMIT k, the number of returned tuples is usually small: k is typically under 100.

There is therefore a tantalizing question: if we do not really care about the order in which the first k answers to a query are returned, can this be used to our advantage? In particular, can this be used to evaluate queries faster? This is precisely the question that the VLDB 2022 paper “Threshold queries in theory and in the wild” set to answer.

At first, it explained a variety of scenarios where such queries appear. These include data exploration (e.g., show me some random 10 tuples from the query answer), cardinality constraint queries (find violations of cardinality restrictions, e.g., limits on sizes of groups), or data monitoring queries (say, connecting a group to a number of related features). Having provided examples of those queries written in SQL, SPARQL, and Cypher, the paper then went ahead with a combination of theoretical and experimental studies.

Theoretical study.

The key contribution here is a clean theoretical model that encompasses the above types of queries and is at the same time amenable to complexity-theoretic and algorithmic analysis.