Chapter VI

Advanced Ad Hoc Star Query Processing

Nikos Karayannidis
National Technical University of Athens, Greece

Aris Tsois
National Technical University of Athens, Greece

Timos Sellis
National Technical University of Athens, Greece

Abstract

Star queries are the most prevalent kind of queries in data warehousing, online analytical processing (OLAP), and business intelligence applications. Thus, there is an imperative need for efficiently processing star queries. To this end, a new class of fact table organizations has emerged that exploits path-based surrogate keys in order to hierarchically cluster the fact table data of a star schema. In the context of these new organizations, star query processing changes radically. In this chapter, we present a complete abstract processing plan that captures all the necessary steps in evaluating such queries over hierarchically clustered fact tables. Furthermore, we realize the abstract operations in terms of physical operations over the CUBE File data structure. Finally we discuss star query optimization issues over the presented abstract plan.
Introduction

Star queries are the most prevalent kind of queries in data warehousing, online analytical processing (OLAP), and business intelligence applications. Star queries impose restrictions on the dimension tables that are used for selecting specific facts from the fact table; these facts are further grouped and aggregated according to the user demands. Furthermore, advanced decision support calls for ad hoc analysis, in contrast to using predefined reports that are constructed periodically, or have already been precomputed. The foundation for this kind of analysis is the support of ad hoc star queries, which comprise the real essence of OLAP. Efficient processing of ad hoc star queries is a very difficult task considering, on one hand, the native complexity of typical OLAP queries, which potentially combine huge amounts of data, and on the other, the fact that no a priori knowledge for the query exists and thus no precomputation of results or other query-specific tuning can be exploited. The only way to evaluate these queries is to access directly the base data in an efficient way.

Traditionally, the major bottleneck in evaluating star queries has been the join of the central (and usually very large) fact table with the surrounding dimension tables (also known as a star join). To cope with this problem various indexing schemes have been developed (Chan & Ioannidis, 1998; O’Neill & Grafe, 1995; O’Neil & Quass, 1997; Sarawagi, 1997; Wu & Buchmann, 1998). Also precomputation of aggregation results has been studied extensively—mainly as a view maintenance problem—and is used as a means of accelerating query performance in data warehouses (Roussopoulos, 1998; Srivastava, Dar, Jagadish & Levy, 1996).

However, for ad hoc star queries the usage of precomputed aggregation results is extremely limited or even impossible in some cases. Even when elaborate indexes are used, due to the arbitrary ordering of the fact table tuples, there might be as many disk page accesses as are the tuples resulting from the fact table. The only alternative one can have for such queries is a good physical clustering of the data, and it is exactly for this reason that a new class of primary organizations for the fact table has emerged (Karayannidis, Sellis, & Kouvaras, 2004; Markl, Ramsak, & Bayern, 1999). These organizations exploit a special kind of key that is based on the hierarchy paths of the dimensions, in order to achieve hierarchical clustering of the facts. This physical clustering results in a reduced I/O cost for the majority of star queries, which are based on the dimension hierarchies. Moreover, in a dimensional data warehouse it is natural to exploit a multidimensional index for storing the tuples. A typical star join is transformed then into a multidimensional range query, which is very efficiently computed using the underlying multidimensional data structures. The combination of the two: hierarchical clustering of data and a multidimensional structure for accessing the fact table tuples results in a very efficient method for ad hoc star query processing.
Intelligent Query Answering
[www.igi-global.com/chapter/intelligent-query-answering/10675?camid=4v1a](www.igi-global.com/chapter/intelligent-query-answering/10675?camid=4v1a)