8. Conclusion and Outlook

Data warehouses are important database applications and will become even more relevant in the next years. The size of the databases and the kind of queries which are processed on data warehouses differ much from transaction-oriented operational systems. The processing of range queries on multidimensional data is especially important for data warehouse systems. The main goal of this thesis is therefore to investigate what kind of index structures support efficiently typical queries in a data warehouse environment. For data warehouse applications multidimensional tree-based index structures and bitmap indexes are the most promising structures. The R^*-tree is well known for its robustness. Variants of the standard bitmap indexing techniques overcome the problem of space inefficiency and the problem of weak support for range queries.

In read-mostly environments, fast query processing is more important than short update phases. The better the data is clustered onto pages, the faster the queries are processed. This thesis described one approach for finding *optimal* index structures by transforming the problem of clustering data into a mixed integer problem (MIP). This approach guarantees finding optimal solutions. Software packages cplex and MOPS can solve the defined MIP numerically. These clusterings are marginally better than the solutions generated by R^*-trees. Due to its time complexity the MIP approach can only be used for very small data sets to evaluate the quality of heuristics. For real size data sets the MIP approach is not applicable. The problem of finding an optimal index structure in polynomial time remains an open research question.

Typical queries in data warehouse applications compute aggregated data from large sets of tuples. For processing this kind of queries, we *improved* tree-based index structures by an extension where materialized aggregated data is stored in the inner nodes. This thesis investigated what kind of data can be calculated and materialized inside the index structure. We showed how to change the insert and query algorithms to maintain and use the aggregated data inside the inner nodes. We presented an upper bound to estimate the space overhead for aggregated materialized data. Experiments showed that the extension decreases significantly the number of necessary disk accesses for processing range queries on aggregated data.

To *estimate* the performance of index structures analytically, we compared performance models for tree-based structures. The GRID model, SUM model, and FRACTAL model are known from literature. The GRID and SUM model assume uniformly distributed data. The FRACTAL model uses the ratio between the part of data space which is covered by the real data and and the whole data space. We extended the models to include aggregated data in the inner nodes of the index structure. Then, we developed the new PISA model (Performance of Index Structures with and without Aggregated data). The main advantage of PISA model compared to the previous existing models is that PISA considers the actual distribution of data and the distribution of queries. In this thesis, we adapted PISA model to uniform, skewed, and normal distributions of data and queries. Experiments showed that the PISA model is more accurate than the other models in most cases.

This thesis provided techniques to *compare* index structures for data warehouses. The performance of index structures depends on different parameters. We described nine parameters which influence query processing time. We defined parameterized performance measures to calculate the expected time needed to execute specified queries. We varied the parameters and generate sets of experimental cases. We generated structured information in form of classification trees from data sets. Classification trees provided concrete rules in which situation which index structure performs best. In addition, classification trees showed which parameters influence the performance of the index structure most. One result of the experiments is that the blocksize does not influence the relative performance between the structures. To see how the performance of index structure depend on certain parameters, an aggregation method condenses high-dimensional data into two-dimensional data. The resulting figures show that bitmap indexes are faster than tree based index structures for at least four dimensions. For less than three dimensions tree structures perform better. Another interesting result is that if the trend of evolving disk technology continues with the same speed as it did in the last years, the bitmap indexes techniques will get more efficient relative to the tree structure with the time. Bitmap indexes profit from the fact that the time gap between a random block access and a sequential block access is getting greater with every new generation of disk technology.

There are many open research questions in this field of index structures for data warehouses. One of the rather theoretically interesting questions is to find an *optimal* index structure in polynomial time.

We believe that storing aggregated data in the inner nodes of a tree structure is sufficiently investigated in this thesis. The presented performance models to estimate the number of disk accesses can be evaluated with more data sets in different experiments. However, techniques to extend the model to different kinds of data and/or to consider the other levels of a tree are presented in this thesis.

An important starting point for research about index structures are bitmap indexing techniques. Bitmap indexes are well suited for high dimensional data with a small number of different values. The bitmap indexes use hardware efficiently because bitmaps indexes read large blocks of data and perform Boolean operations on these large blocks of bits. Therefore, bitmap indexes exhibit advantages over tree-based indexing techniques for many data warehouse applications. Since they profit from new disk technology more than tree-based indexing methods and they are not investigated in such detail as the tree-based index structures are, we believe that there is a great potential for developing new indexing structures based on bitmaps.