INTRODUCTION

Scientific databases and data warehouses store large amounts of data in several tables and attributes. For instance, the Sloan Digital Sky Survey (SDSS) astronomical database contains a large number of tables with hundreds of attributes, which can be queried in various combinations (Papadomanolakis & Ailamaki, 2004). These queries involve many tables using binary operations, such as joins. To speed up these queries, many optimization structures were proposed that can be divided into two main categories: redundant structures like materialized views, advanced indexing schemes (bitmap, bitmap join indexes, etc.) (Sanjay, Chaudhuri & Narasayya, 2000) and vertical partitioning (Sanjay, Narasayya & Yang 2004) and non redundant structures like horizontal partitioning (Sanjay, Narasayya & Yang 2004; Bellatreche, Boukhalfa & Mohania, 2007) and parallel processing (Datta, Moon, & Thomas, 2000; Stöhr, Märtens & Rahm, 2000). These optimization techniques are used either in a sequential manner or combined. These combinations are done in two main categories: redundant structures like materialized views, advanced indexing schemes (bitmap, bitmap join indexes, etc.) (Sanjay, Chaudhuri & Narasayya, 2000) and vertical partitioning (Sanjay, Narasayya & Yang 2004) and non redundant structures like horizontal partitioning (Sanjay, Narasayya & Yang 2004; Bellatreche, Boukhalfa & Mohania, 2007) and parallel processing (Datta, Moon, & Thomas, 2000; Stöhr, Märtens & Rahm, 2000). These optimization techniques are used either in a sequential manner or combined. These combinations are done in two main categories: redundant structures like materialized views, advanced indexing schemes (bitmap, bitmap join indexes, etc.) (Sanjay, Chaudhuri & Narasayya, 2000) and vertical partitioning (Sanjay, Narasayya & Yang 2004) and non redundant structures like horizontal partitioning (Sanjay, Narasayya & Yang 2004; Bellatreche, Boukhalfa & Mohania, 2007) and parallel processing (Datta, Moon, & Thomas, 2000; Stöhr, Märtens & Rahm, 2000). These optimization techniques are used either in a sequential manner or combined. These combinations are done in two main categories: redundant structures like materialized views, advanced indexing schemes (bitmap, bitmap join indexes, etc.) (Sanjay, Chaudhuri & Narasayya, 2000) and vertical partitioning (Sanjay, Narasayya & Yang 2004) and non redundant structures like horizontal partitioning (Sanjay, Narasayya & Yang 2004; Bellatreche, Boukhalfa & Mohania, 2007) and parallel processing (Datta, Moon, & Thomas, 2000; Stöhr, Märtens & Rahm, 2000). These optimization techniques are used either in a sequential manner or combined. These combinations are done in two main categories: redundant structures like materialized views, advanced indexing schemes (bitmap, bitmap join indexes, etc.) (Sanjay, Chaudhuri & Narasayya, 2000) and vertical partitioning (Sanjay, Narasayya & Yang 2004) and non redundant structures like horizontal partitioning (Sanjay, Narasayya & Yang 2004; Bellatreche, Boukhalfa & Mohania, 2007) and parallel processing (Datta, Moon, & Thomas, 2000; Stöhr, Märtens & Rahm, 2000). These optimization techniques are used either in a sequential manner or combined. These combinations are done in two main categories: redundant structures like materialized views, advanced indexing schemes (bitmap, bitmap join indexes, etc.) (Sanjay, Chaudhuri & Narasayya, 2000) and vertical partitioning (Sanjay, Narasayya & Yang 2004) and non redundant structures like horizontal partitioning (Sanjay, Narasayya & Yang 2004; Bellatreche, Boukhalfa & Mohania, 2007) and parallel processing (Datta, Moon, & Thomas, 2000; Stöhr, Märtens & Rahm, 2000). These optimization techniques are used either in a sequential manner or combined. These combinations are done in two main categories: redundant structures like materialized views, advanced indexing schemes (bitmap, bitmap join indexes, etc.) (Sanjay, Chaudhuri & Narasayya, 2000) and vertical partitioning (Sanjay, Narasayya & Yang 2004) and non redundant structures like horizontal partitioning (Sanjay, Narasayya & Yang 2004; Bellatreche, Boukhalfa & Mohania, 2007) and parallel processing (Datta, Moon, & Thomas, 2000; Stöhr, Märtens & Rahm, 2000).
are two similar optimization techniques - both speed up query execution, pre-compute join operations and concern selection attributes of dimension tables. Furthermore, BJIs and HP can interact with one another, i.e., the presence of an index can make a partitioned schema more efficient and vice versa (since fragments have the same schema of the global table, they can be indexed using BJIs and BJIs can also be partitioned (Sanjay, Narasayya & Yang 2004)).

**BACKGROUND**

Note that each BJI can be defined on one or several non key dimension’s attributes with a low cardinality (that we call indexed columns) by joining dimension tables owned these attributes and the fact table.

**Definition:** An indexed attribute $A_i$ candidate for defining a BJI is a column $A_i$ of a dimension table $D_i$ with a low cardinality (like gender attribute) such that there is a selection predicate of the form: $D_i.A_i \theta$ value, $\theta$ is one of six comparison operators {$=,<,>,\leq,\geq$}, and value is the predicate constant.

For a large number of indexed attributes candidates, selecting optimal BJIs is an NP-hard problem (Bellatreche, Boukhalfa & Mohania, 2007).

On the other hand, the best way to partition a relational data warehouse is to decompose the fact table based on the fragmentation schemas of dimension tables (Bellatreche & Boukhalfa, 2005). Concretely, (1) partition some/all dimension tables using their simple selection predicates $(D_i.A_i \theta$ value), and then (2) partition the facts table using the fragmentation schemas of the fragmented dimension tables (this fragmentation is called derived horizontal fragmentation (Özsu a Valduriez, 1999)). This fragmentation procedure takes into consideration the star join queries requirements.

The number of horizontal fragments (denoted by $N$) of the fact table generated by this partitioning procedure is given by:

$$N = \prod_{i=1}^{g} m_i,$$

where $m_i$ and $g$ are the number of fragments of the dimension table $D_i$ and the number of dimension tables participating in the fragmentation process, respectively. This number may be very large (Bellatreche & Boukhalfa & Abdalla, 2006). Based on this definition, there is a strong similarity between BJIs and horizontal partitioning as show the next section.

**Similarity between HP and BJIs**

To show the similarity between HP and BJIs, the following scenario is considered. Suppose a data warehouse represented by three dimension tables (TIME, CUSTOMER and PRODUCT) and one fact table (SALES). The population of this schema is given in Figure 1. On the top of this the following query is executed:

```sql
SELECT Count(*)
FROM CUSTOMER C, PRODUCT P, TIME T, SALES S
WHERE C.City='LA'
AND P.Range='Beauty'
AND T.Month='June'
AND P.PID=S.PID
AND C.CID=S.CID
AND T.TID=S.TID
```

This query has three selection predicates defined on dimension table attributes City (City='LA'), Range (Range='Beauty') and Month (Month = 'June') and

**Figure 1. Sample of data warehouse population**
Related Content

VRMiner: A Tool for Multimedia Database Mining With Virtual Reality
[www.irma-international.org/chapter/vrminer-tool-multimedia-database-mining/7715/](www.irma-international.org/chapter/vrminer-tool-multimedia-database-mining/7715/)

Data Mining in Security Applications
[www.irma-international.org/chapter/data-mining-security-applications/10863/](www.irma-international.org/chapter/data-mining-security-applications/10863/)

Learning Temporal Information from Text
[www.irma-international.org/chapter/learning-temporal-information-text/10966/](www.irma-international.org/chapter/learning-temporal-information-text/10966/)

Active Disks for Data Mining
Alexander Thomasian (2005). *Encyclopedia of Data Warehousing and Mining* (pp. 6-11).
[www.irma-international.org/chapter/active-disks-data-mining/10556/](www.irma-international.org/chapter/active-disks-data-mining/10556/)

Flexible Mining of Association Rules
[www.irma-international.org/chapter/flexible-mining-association-rules/10925/](www.irma-international.org/chapter/flexible-mining-association-rules/10925/)