Conceptual models are useful to design database applications since they favor the communication between the stakeholders in a project. However, conceptual models must be translated into logical ones for their implementation on a database management system. In this chapter, we study how the conceptual multidimensional model studied in the previous chapter can be represented in the relational model. We start in Sect. 5.1 by describing the three logical models for data warehouses, namely, relational OLAP (ROLAP), multidimensional OLAP (MOLAP), and hybrid OLAP (HOLAP). In Sect. 5.2, we focus on the relational representation of data warehouses and study four typical implementations: the star, snowflake, starflake, and constellation schemas. In Sect. 5.3, we present the rules for mapping a conceptual multidimensional model (in our case, the MultiDim model) to the relational model. Section 5.4 discusses how to represent the time dimension. Sections 5.5 and 5.6 study how hierarchies, facts with multiple granularities, and many-to-many dimensions can be implemented in the relational model. Section 5.7 is devoted to the study of slowly changing dimensions, which arise when dimensions in a data warehouse are updated. In Sect. 5.8, we study how a data cube can be represented in the relational model and how it can be queried in SQL using the SQL/OLAP extension. Finally, to show how these concepts are applied in practice, in Sects. 5.9 and 5.10, we show how the Northwind cube can be implemented, respectively, in Microsoft Analysis Services and in Mondrian.

5.1 Logical Modeling of Data Warehouses

There are several approaches for implementing a multidimensional model, depending on how the data cube is stored. These approaches are:
• **Relational OLAP (ROLAP)**, which stores data in relational databases and supports extensions to SQL and special access methods to efficiently implement the multidimensional data model and the related operations.

• **Multidimensional OLAP (MOLAP)**, which stores data in specialized multidimensional data structures (e.g., arrays) and implements the OLAP operations over those data structures.

• **Hybrid OLAP (HOLAP)**, which combines both approaches.

In ROLAP systems, multidimensional data are stored in relational tables. Further, in order to increase performance, aggregates are also precomputed in relational tables (we will study aggregate computation in Chap. 7). These aggregates, together with indexing structures, take a large space from the database. Moreover, since multidimensional data reside in relational tables, OLAP operations must be performed on such tables, yielding usually complex SQL statements. Finally, in ROLAP systems, all data management relies on the underlying relational DBMS. This has several advantages since relational databases are well standardized and provide a large storage capacity.

In MOLAP systems, data cubes are stored in multidimensional arrays, combined with hashing and indexing techniques. Therefore, the OLAP operations can be implemented efficiently, since such operations are very natural and simple to perform. Data management in MOLAP is performed by the multidimensional engine, which generally provides less storage capacity than ROLAP systems. Normally, typical index structures (e.g., B-trees, or R-trees) are used to index sparse dimensions (e.g., a product or a store dimension), and dense dimensions (like the time dimension) are stored in lists of multidimensional arrays. Each leaf node of the index tree points to such arrays, providing efficient cube querying and storage, since the index in general fits in main memory. Normally, MOLAP systems are used to query data marts where the number of dimensions is relatively small (less than ten, as a popular rule of thumb). For high-dimensionality data, ROLAP systems are used. Finally, MOLAP systems are proprietary, which reduces portability.

While MOLAP systems offer less storage capacity than ROLAP systems, they provide better performance when multidimensional data are queried or aggregated. Thus, HOLAP systems benefit from the storage capacity of ROLAP and the processing capabilities of MOLAP. For example, a HOLAP server may store large volumes of detailed data in a relational database, while aggregations are kept in a separate MOLAP store.

Current OLAP tools support a combination of the above models. Nevertheless, most of these tools rely on an underlying data warehouse implemented on a relational database management system. For this reason, in what follows, we study the relational OLAP implementation in detail.