Introduction to Databases and Data Warehouses

This chapter introduces the basic concepts of databases and data warehouses. It compares the two fields and stresses the differences and complementarities between them. The aim of this chapter is to define the terminology and the framework used in the rest of the book, not to provide an extensive coverage of these fields. The outline of this chapter is as follows.

The first part of the chapter is devoted to databases. A database is a collection of logically related data that supports the activities of an organization. Many of the activities of our everyday life involve some interaction with a database. Section 2.1 begins by describing the basic concepts underlying database systems and describes the typical four-step process used for designing them, starting with requirements specification, and followed by conceptual, logical, and physical design. These steps allow a separation of concerns, where requirements specification gathers the requirements about the application and its environment, conceptual design targets the modeling of these requirements from the perspective of the users, logical design develops an implementation of the application according to a particular database technology, and physical design optimizes the application with respect to a particular implementation platform. We review in Sect. 2.2 the entity-relationship model, a popular conceptual model for designing databases. Section 2.3 is devoted to two logical models of databases, the relational and the object-relational model. Finally, physical design considerations for databases are covered in Sect. 2.4.

The second part of the chapter is devoted to data warehouses. A data warehouse is a particular database targeted toward decision support. It takes data from various operational databases and other data sources and transforms it into new structures that fit better for the task of performing business analysis. We review in Sect. 2.5 the basic characteristics of data warehouses and compare them with operational databases. As explained in Sect. 2.6, data warehouses are based on a multidimensional model, where data is represented as hypercubes with dimensions corresponding to the various business perspectives and cube cells containing the measures to be analyzed. Sections 2.7 and 2.8 cover, respectively, logical and physical data warehouse design. The archi-
2 Introduction to Databases and Data Warehouses

The architecture of data warehouse systems is described in detail in Sect. 2.9; as we shall see, in addition to the data warehouse itself, data warehouse systems are composed of back-end tools, which extract data from the various sources to populate the warehouse, and front-end tools, which are used to extract the information from the warehouse and present it to users. Sections 2.10 and 2.11 briefly present two representative data warehouse tools, SQL Server Analysis Services 2005 and Oracle OLAP 10g. Finally, Sect. 2.12 concludes this chapter.

2.1 Database Concepts

Databases constitute the core component of today’s information systems. A **database** is a shared collection of logically related data, and a description of that data, designed to meet the information needs and support the activities of an organization. A database is deployed on a **database management system** (DBMS), which is a software system that allows users to define, create, manipulate, and administer a database.

Designing a database system is a complex undertaking that is typically divided into four phases:

- **Requirements specification**, which collects information about the users’ needs with respect to the database system. A large number of approaches for requirements specification have been developed by both academia and practitioners. In general, these techniques help to elicit necessary and desirable system properties from prospective users and/or project managers, to homogenize requirements, and to assign priorities to them, i.e., separate necessary from “nice to have” system properties [318]. During this phase, active participation of users will increase customer satisfaction with the delivered system and avoid errors, which can be very expensive to correct if the subsequent phases have already been developed.

- **Conceptual design**, which aims at building a user-oriented representation of the database that does not contain any implementation considerations. This is done by using a **conceptual model** in order to identify the relevant entities, relationships, and attributes of the application domain. The entity-relationship model is one of the most often used conceptual models for designing database applications. Alternatively, object-oriented modeling techniques can also be applied, based on UML notation [30].

  Conceptual design can be performed using two different approaches, according to the complexity of the system and the developers’ experience:

  - **Top-down design**: the requirements of the various users are merged before the design process begins, and a unique schema is built. Afterwards, a separation of the views corresponding to individual users’