**Chapter 5 Physical Database Design and Performance**

**Chapter Overview**

This chapter presents the basic steps that are required to develop an effective physical database design. Physical database design is very important as it immediately affects factors that are important to the end user and the entire enterprise: data integrity and security, response times, usability and user experience, and so on. First, we present a simple approach to estimating the volume of data in a database, as well as the probable data usage patterns. Next, we discuss issues associated with defining fields, including data type determination, coding and compression techniques, controlling data integrity, and handling missing data. We then discuss designing physical records and include an expanded section on denormalization. Vertical and horizontal partitioning are covered next. We describe the basic file organizations and the trade-offs that are typically involved in selecting a file organization. We examine the use and types of indexes. File access performance is discussed, including a discussion of query optimization. Overall, the chapter emphasizes the physical design process and the goals of that process.

**Chapter Objectives**

1. Present physical database design as a critical element in achieving overall database objectives, rather than as an afterthought.
2. Ensure that students understand the factors that must be considered in distributing data effectively and how a simple model can be used to obtain at least a first-cut distribution.
3. Provide students with a sound understanding of the use of indexes and the trade-offs that must be considered in their use.
4. Ensure students understand that denormalization must be used with great care and for specific reasons.

**Key Terms**

|  |  |  |
| --- | --- | --- |
| Data type | Hashed file organization | Physical file |
| Denormalization | Hashing algorithm | Pointer |
| Extent | Horizontal partitioning | Secondary key |
| Field | Index | Sequential file organization |
| File organization | Indexed file organization | Tablespace |
| Hash index table | Join index | Vertical partitioning |