- Free list
- Variable-length records
  - Slotted-page structure
- Large objects
- Heap file organization
- Sequential file organization
- Hashing file organization
- Multitable clustering file organization
- Search key
- Data dictionary
- System catalog
- Buffer
  - Buffer manager
  - Pinned blocks
  - Forced output of blocks
- Buffer-replacement policies
  - Least recently used (LRU)
  - Toss-immediate
  - Most recently used (MRU)

Practice Exercises

10.1 Consider the data and parity-block arrangement on four disks depicted in Figure 10.18. The B_is represent data blocks; the P_is represent parity blocks. Parity block P_i is the parity block for data blocks B_{4i-3} to B_{4i}. What, if any, problem might this arrangement present?

10.2 Flash storage:

a. How is the flash translation table, which is used to map logical page numbers to physical page numbers, created in memory?

b. Suppose you have a 64 gigabyte flash storage system, with a 4096 byte page size. How big would the flash translation table be, assuming each page has a 32 bit address, and the table is stored as an array.

c. Suppose how to reduce the size of the translation table if very often long ranges of consecutive logical page numbers are mapped to consecutive physical page numbers.

Disk 1	Disk 2	Disk 3	Disk 4
B_1	B_2	B_3	B_4
P_1	B_5	B_6	B_7
B_8	P_2	B_9	B_{10}
⋮	⋮	⋮	⋮

Figure 10.18 Data and parity block arrangement.

10.3 A power failure that occurs while a disk block is being written could result in the block being only partially written. Assume that partially written blocks can be detected. An atomic block write is one where either the disk block is fully written or nothing is written (i.e., there are no partial writes). Suggest schemes for getting the effect of atomic block writes with the following RAID schemes. Your schemes should involve work on recovery from failure.

a. RAID level 1 (mirroring)

b. RAID level 5 (block interleaved, distributed parity)

10.4 Consider the deletion of record 5 from the file of Figure 10.6. Compare the relative merits of the following techniques for implementing the deletion:

a. Move record 6 to the space occupied by record 5, and move record 7 to the space occupied by record 6.

b. Move record 7 to the space occupied by record 5.

c. Mark record 5 as deleted, and move no records.

10.5 Show the structure of the file of Figure 10.7 after each of the following steps:

a. Insert (24556, Turnamian, Finance, 98000).

b. Delete record 2.

c. Insert (34556, Thompson, Music, 67000).

10.6 Consider the relations *section* and *takes*. Give an example instance of these two relations, with three sections, each of which has five students. Give a file structure of these relations that uses multitable clustering.

10.7 Consider the following bitmap technique for tracking free space in a file. For each block in the file, two bits are maintained in the bitmap. If the block is between 0 and 30 percent full the bits are 00, between 30 and 60 percent the bits are 01, between 60 and 90 percent the bits are 10, and above 90 percent the bits are 11. Such bitmaps can be kept in memory even for quite large files.

a. Describe how to keep the bitmap up to date on record insertions and deletions.

b. Outline the benefit of the bitmap technique over free lists in searching for free space and in updating free space information.

10.8 It is important to be able to quickly find out if a block is present in the buffer, and if so where in the buffer it resides. Given that database buffer sizes are very large, what (in-memory) data structure would you use for the above task?

10.9 Give an example of a relational-algebra expression and a query-processing strategy in each of the following situations:

a. MRU is preferable to LRU.

b. LRU is preferable to MRU.

Exercises

10.10 List the physical storage media available on the computers you use routinely. Give the speed with which data can be accessed on each medium.

10.11 How does the remapping of bad sectors by disk controllers affect data-retrieval rates?

10.12 RAID systems typically allow you to replace failed disks without stopping access to the system. Thus, the data in the failed disk must be rebuilt and written to the replacement disk while the system is in operation. Which of the RAID levels yields the least amount of interference between the rebuild and ongoing disk accesses? Explain your answer.

10.13 What is scrubbing, in the context of RAID systems, and why is scrubbing important?

10.14 In the variable-length record representation, a null bitmap is used to indicate if an attribute has the null value.

a. For variable length fields, if the value is null, what would be stored in the offset and length fields?

b. In some applications, tuples have a very large number of attributes, most of which are null. Can you modify the record representation such that the only overhead for a null attribute is the single bit in the null bitmap.

10.15 Explain why the allocation of records to blocks affects database-system performance significantly.

10.16 If possible, determine the buffer-management strategy used by the operating system running on your local computer system and what mechanisms it provides to control replacement of pages. Discuss how the control on replacement that it provides would be useful for the implementation of database systems.

10.17 List two advantages and two disadvantages of each of the following strategies for storing a relational database:

a. Store each relation in one file.

b. Store multiple relations (perhaps even the entire database) in one file.

10.18 In the sequential file organization, why is an overflow *block* used even if there is, at the moment, only one overflow record?

10.19 Give a normalized version of the *Index_metadata* relation, and explain why using the normalized version would result in worse performance.

10.20 If you have data that should not be lost on disk failure, and the data are write intensive, how would you store the data?

10.21 In earlier generation disks the number of sectors per track was the same across all tracks. Current generation disks have more sectors per track on outer tracks, and fewer sectors per track on inner tracks (since they are shorter in length). What is the effect of such a change on each of the three main indicators of disk speed?

10.22 Standard buffer managers assume each block is of the same size and costs the same to read. Consider a buffer manager that, instead of LRU, uses the rate of reference to objects, that is, how often an object has been accessed in the last n seconds. Suppose we want to store in the buffer objects of varying sizes, and varying read costs (such as Web pages, whose read cost depends on the site from which they are fetched). Suggest how a buffer manager may choose which block to evict from the buffer.

Bibliographical Notes

Hennessy et al. [2006] is a popular textbook on computer architecture, which includes coverage of hardware aspects of translation look-aside buffers, caches, and memory-management units. Rosch [2003] presents an excellent overview of computer hardware, including extensive coverage of all types of storage technology such as magnetic disks, optical disks, tapes, and storage interfaces. Patterson [2004] provides a good discussion on how latency improvements have lagged behind bandwidth (transfer rate) improvements.

With the rapid increase in CPU speeds, cache memory located along with the CPU has become much faster than main memory. Although database systems do not control what data is kept in cache, there is an increasing motivation to organize data in memory and write programs in such a way that cache utilization is maximized. Work in this area includes Rao and Ross [2000], Ailamaki et al. [2001], Zhou and Ross [2004], Garcia and Korth [2005], and Cieslewicz et al. [2009].

The specifications of current-generation disk drives can be obtained from the Web sites of their manufacturers, such as Hitachi, LaCie, Iomega, Seagate, Maxtor, and Western Digital.

Discussions of redundant arrays of inexpensive disks (RAID) are presented by Patterson et al. [1988]. Chen et al. [1994] presents an excellent survey of RAID principles and implementation. Reed–Solomon codes are covered in Pless [1998].

Buffering data in mobile systems is discussed in Imielinski and Badrinath [1994], Imielinski and Korth [1996], and Chandrasekaran et al. [2003].

The storage structure of specific database systems, such as IBM DB2, Oracle, Microsoft SQL Server, and PostgreSQL are documented in their respective system manuals.

Buffer management is discussed in most operating-system texts, including in Silberschatz et al. [2008]. Chou and Dewitt [1985] presents algorithms for buffer management in database systems, and describes a performance evaluation.