



## 12.1

SSDs can be used as a storage layer between memory and magnetic disks, with some parts of the database (e.g., some relations) stored on SSDs and the rest on magnetic disks. Alternatively, SSDs can be used as a buffer or cache for magnetic disks; frequently used blocks would reside on the SSD layer, while infrequently used blocks would reside on magnetic disks.

a. Which of the two alternatives would you choose if you need to support real-time queries that must be answered within a guaranteed short period of time? Explain why.

b. Which of the two alternatives would you choose if you had a very large customer relation, where only some disk blocks of the relation are accessed frequently, with other blocks rarely accessed.

- a. 我会选择将 SSD 用作缓存，因为 SSD 的读写速度比磁盘快得多，可以更快地响应查询。
- b. 我会选择将 SSD 用作存储层，因为只有一些磁盘块经常被访问，其他块很少被访问，SSD 的读写速度比磁盘快得多，可以更快地访问经常访问的块。

## 13.5

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 this task?

我会使用哈希表，因为哈希表的查找时间复杂度平均是  $O(1)$ ，符合要求。

## 13.9

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?

a.

- 偏移字段：任何数值都可以，因为长度字段的特殊值（如  $-1$ ）表示字段为空。
- 长度字段：使用特殊值（如  $-1$ ）来表示该字段为空，不占用任何实际存储空间。

b. 可以。

在记录开头添加一个空位图（Null Bitmap）字段，这样我们可以仅用一位来表示每个属性是否为空——空属性不在偏移列表中出现，以此避免额外存储开销。

## 13.11

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.

a.

- 优势：
  - i. 读取速度快：由于每个文件只包含一个关系，读取一个完整关系的数据时无需扫描不相关的关系，提高了读取效率。
  - ii. 管理简单：直接访问单一文件，数据组织清晰，管理容易。
- 劣势：
  - i. 更新效率低：对于大型关系的更新需要重新写入整个文件，导致操作开销较高。
  - ii. 空间浪费：如果一个关系的数据量很小，单独为其分配文件可能会导致文件系统中产生大量小文件，浪费磁盘空间。

b.

- 优势：
  - i. 空间利用率高：通过共享文件空间，减少了因小关系单独存储而浪费的磁盘空间。
  - ii. 文件数量少：减少文件系统中文件的数量，有助于更好地组织和管理数据。
- 劣势：

- i. **读取速度慢：** 读取单个关系时需要扫描整个文件，可能会影响读取效率，尤其是当文件非常大时。
- ii. **复杂性增加：** 必须使用额外的元数据来定位各个关系的位置和长度，增加了数据组织和管理复杂性。