

CMU 15-445 - Database Systems Lecture 3 - Database Storage 1

Database Storage 1

Memory Hierarchy

We will assume the database management system will have to deal with 2 levels of memory - volatile and non-volatile. This lecture and the next few lectures are focussed on how the data is actually stored in the file system.

File Storage

We want to give the user the impression that the entire database is stored in the main memory, and not stall when the user requests data from outside the main memory, even though calls to disk are slow.

We want to implement something like a virtual memory.

Now, the operating system can already do this well, but we almost never want to leave it up to the OS to handle this for us, because we can manage memory much better since we have extra information about the queries that are being run and that can be run, and we can use that information to perform spatial and temporal optimizations. Moreover, some queries may require specialized memory management for correctness.

A DBMS stores a database in memory as a file (sqlite) or multiple files. The OS does not know anything about what is inside those files, only the DBMS can interpret them correctly.

The DBMS has a *storage manager* that is responsible for managing these files. It stores the database in these files in blocks of data called pages. A page is just a block of data that can contain any kind of information. There are three kinds of pages -

1. Hardware pages - The blocks exposed by the SSD/disk itself on the hardware level. Usually 4kB.
2. OS page - The blocks as defined by the OS (4kB).

3. Database page - The blocks as defined by the DBMS (0.5 to 16kB).

Heap File Organization

There are a few ways to organize pages in memory, and the one covered in the lecture is the heap file organization. A heap file just means that pages are stored in memory in an unordered way, and tuples are stored inside those pages.

We maintain something called a page directory, which is basically a hash table that maps page ids to the location of the pages in memory. It can also contain metadata about the page.

Page Structure

The data contained in the page itself can be stored in two ways - slotted pages and log-structured pages.

Slotted pages are pages which contain “slots” for tuples to go into. The page has a header that stores metadata about the page, and the top of the page contains a “slots” array that maps slot locations to an offset in the page. The tuples are stored starting from the bottom of the page working up. The page is said to be full when the slots array and the tuples touch.

Log structured pages store only logs, i.e, records of how the page was modified. The data present in the page can then be inferred from the logs during a read. This results in fast writes, but could lead to slow reads. However, there are a few techniques to make reads practical as well.