Summary - CMU lec3

* Disk manager interact with files system.
* DBMS assumes that the primary storage location of the DB is on non-volatile disk .
* DBMS manage the movement of data between volatile and non-volatile storage .
* Storage hierarchy started with network storage which is “ slower , larger and cheaper “ , HDD , SSD , DRAM ,CPU caches and CPU register which is “ faster , smaller and expensive “ .
* Random access slower than sequential .
* DBMS will want to maximize sequential access .
* System design goals :

1. Allow DBMS to manage DB that exceed the amount of memory available .
2. Reading / Writing to disk expensive , so it must be managed carefully .

3 - Random access slower than sequential and DBMS will want to maximize sequential access .

* DBMS can use memory mapping to store the content of a file into address space of a program .
* OS is responsible for moving the pages of the file in and out of memory .
* If you allow multiple the reads to access the mmap , this works good enough for read-only .
* To use the os you used some solutions like

:

1-mlock

2-msync

3-madvise

* DBMS want to control things itself and can do a better job than os .
* Storage manager is responsible for maintining a DB files .
* Page is fixed-size block of data .
* Most systems don’t mix page types .
* Each page is given a unique identifier .
* HW pages is largest block of data that the storage devic can guarantee fail safe writes

.

* Heap file an unordered collection of pages with tuples that are stored in random ordered have two types :

1. Linked list
2. page directory

\* Linked list maintain a header page at the beginning of the file that stores two pointers : 1- Head of the free page list .

2- Head of the data page list .

* Every page contains hearders of metadata the pages’s contents.
* Page layout for any pages storage architecture , we now need to decide how to organize the data inside the page .
* Most common layout scheme is called slotted pages .
* DBMS need away to keep track of individual tuples and each one is unique record identifier .
* Each tuple is prefixed with a header that contains meta-data about it .
* Tuple is essentially a sequence of bytes .
* DBMS can physically denormalize related tuples and store them together in the same page .
* Several No SQL DBMSs do this without calling it physical denormalization .