Hello, in our journey to learn anything we interested in learn only the function of this thing or the fast way to achieve it, we take care about crusts, BUT in science and academic methods you care and learn the basic of anything you want to learn SO, our discussing now about Database in low level(disk and memory).

First, you should know the DBMS assume the primary storage location of the database is non-volatile disk.

Then, I want to show the different between two important concepts:

Random access: which is almost used in memory (volatile) because it's space mainly small .

Sequential access: which is almost used in disk (non-volatile) because it's area mainly large .

SO, DBMS will want to maximize sequential access.

NOW, when we said low level or hardware we obvious think in Operating System

SO, why we don't use OS to manage database in low level?

It's logical thinking, DBMS can use memory mapping (tool of OS) to store a content of a file into the address space in the program, and OS is responsible for

Moving the pages of the file in and out of memory, so, the DBMS doesn't need to worry about it.

BUT, why not use OS?

What if we allow multiple threads to access the MMAP files to hide page faults stalls ?

This work good enough for read only access, it's complicated when there are multiple writers.

Memory mapped problems:

1-Transaction safety:

OS can flush dirty pages at any time.

2-I/O stalls:

DBMS doesn't know which pages are in memory, OS will stall a thread on

a page fault.

3-Error handling

4-Performance Issues:

OS data structures connections, TLB shoot downs.

SO, the idea that DBMS always want to control things itself and can do better than OS.

Database Storage:

We will discuss three components of storage.

1- File storage:
The DBMS stores a database as one or more files on disk typically in a proprietary format
Note – storage manager is responsible for maintain a database files and scheduling for reads and writes.

2-Page storage:

There are three different notions of pages in DBMS:

1-Hardware pages (4KB).

2-OS pages (4KB).

3-Database page (512B - 16KB) .

Page storage Architecture:

Different DBMS manage pages in files on disk in different ways:

Heap, tree or hashing.

Heap file:

A heap file is unordered collection of pages with tuples that are stored in random order.

Create / Get / Write / Delete page.

Must also support iterative over all pages.

3-Tuples storage:

Strawman idea:

Keep track of the number of the tuples in a page and then append a new Tuple at the end.

But it has drawbacks:

What happen if we delete a tuple.

What happen if we have a variable length attribute.

SO, we used the most common layout scheme is called Slotted pages, which slot array maps " slots " to the tuples starting position offset.

The header keeps track of :

1- The number of used slots.

The offset of the starting location of the last slot used.