**Disk oriented database: This basically mean that the software Is going to be designed to assume that the database resides the primary storage location of Database is on some kind. of non-volatile storage (non volatile disk)**

**Storage hierarchy**

**volatile RANDOM Access Byte Addressable CPU Registers.**

**1) Cpu**

**2) Caches**

**3)DRAM**

**These are consider a memory**

**- Faster, smaller and expensive**

**• non volatile sequential Access Block addressable:-**

**(1)SSD**

**(2)HDD**

**(3) Network storage**

**-Slower, cheaper**

**Random access means: Storage I can jump to any offset in the device and get it with roughly. same speed as any other offset and by that mean it's quite addressable**

**-anything below that will be non volatile**

**to go get a block or the page that Contain the data that i want and even though i may only 1001 want bu bits, i gott get the whole things, in some cases to these**

**the speed in which i can access**

**data from these non-Volatile**

**devices will be faster if i'm doing sequential access.**

**algorithms try to reduce number of writes to random pages so that data is stored in Contigous blocks**

**Sequential vs Random access**

**Random access on non-volatile storage is almost always much slower than sequential access.**

**DBMS will want to maximize sequential access.**

**→ Algorithms try to reduce number of writes to random pages so that data is stored in contiguous blocks.**

**Allocating multiple pages at the same time is called an extent.**

**and because the sequential access is better we choose algorithms or methods that will maximize the Quantum**

**- so we have some kind of non-volatile storage and then we have database file and the the database File is going to be broken up into pages or blocks above division line:- -and we have memory and this is**

**going to have a buffer manager that can have some to cation in memory where it can copy some pages in and out it also call page caches**

**SYSTEM DESIGN GOALS**

**Allow the DBMS to manage databases that exceed the amount of memory available.**

**Reading/writing to disk is expensive, so it must be managed carefully to avoid large stalls and performance degradation.**

**Random access on disk is usually much slower than sequential access, so the DBMS will want to maximize sequential access.**

**WHY NOT USE THE OS?**

**DBMS (almost) always wants to control things itself and can do a better job than the OS.**

**→Flushing dirty pages to disk in the correct order.**

**→Specialized prefetching.**

**→Buffer replacement policy.**

**→Thread/process scheduling.**

**The OS is not your friend.**

**basically, memory managed by DB system to move Pages and out From disK**

**•storage manger responsible For maintaining a databases file and it Figure's out where the Files are located what's the directory.**

**Thierarchy is Some will have their dispatcher. So that when a thread inside the**

**DB System or process inside the DB System goes and reads a page. it can then schedule that for you or you can let the OS do the read For you.**

**FILE STORAGE**

**The DBMS stores a database as one or more files on disk typically in a proprietary format.**

**→ The OS doesn't know anything about the contents of these files.**

**Early systems in the 1980s used custom file systems on raw storage.**

**Some "enterprise" DBMSs still support this.**

**Most newer DBMSs do not do thi**s.