Clustered Index Organization

Since says how records in file ordered can only have one on a table.
Useful criteria in determining a/c field should be clustered indexed is to try to guess the likelihood index-only evaluation. That is, the data record itself doesn't need to be accessed only the index.
Ex) Aggregate on age where age clustered

Composite Search Keys

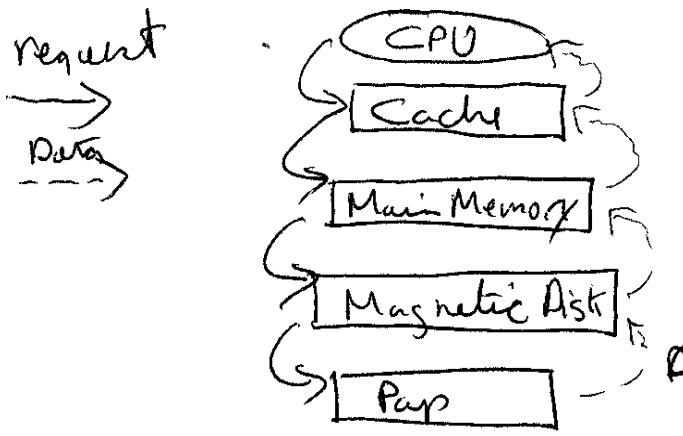
Can have ~~keys~~ index on more than one field: (age, sal)
Equality query → look for exact match i.e. age = 10, sal = 5000
Range query - range of values age > 50, sal < 100000
~~Composite~~
Composite keys support broader range of queries but need to be updated more
Use tree indices rather than hash for range queries
Can be useful for aggregate queries since increase odds of index only scan

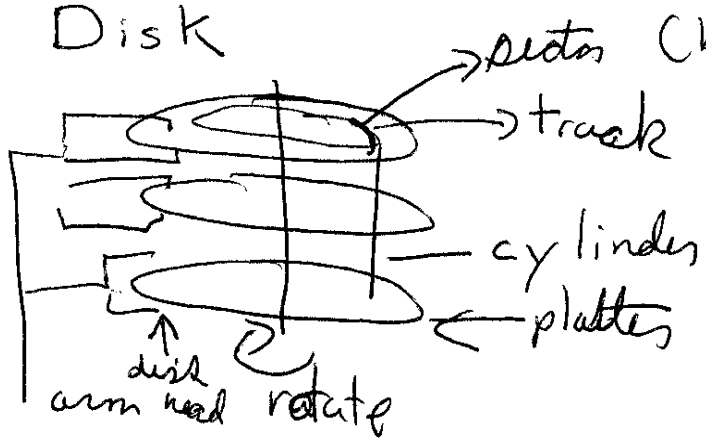
SQL for Creating Indexes

```
CREATE INDEX IndAgeRating ON Students
    WITH STRUCTURE = BTREE
    KEY = (age, gpa)
```

More on Storage Ch 9

Memory Hierarchy





Disk controller provides interface to computer

Checksum used to ensure read and write of blocks okay.

Comparison of disk access verses main memory

Disk access	10mS	~200,000 times
Main memory	50nS	

What makes up a disk access?

Seek Time to find track

Rotational delay

Transfer time