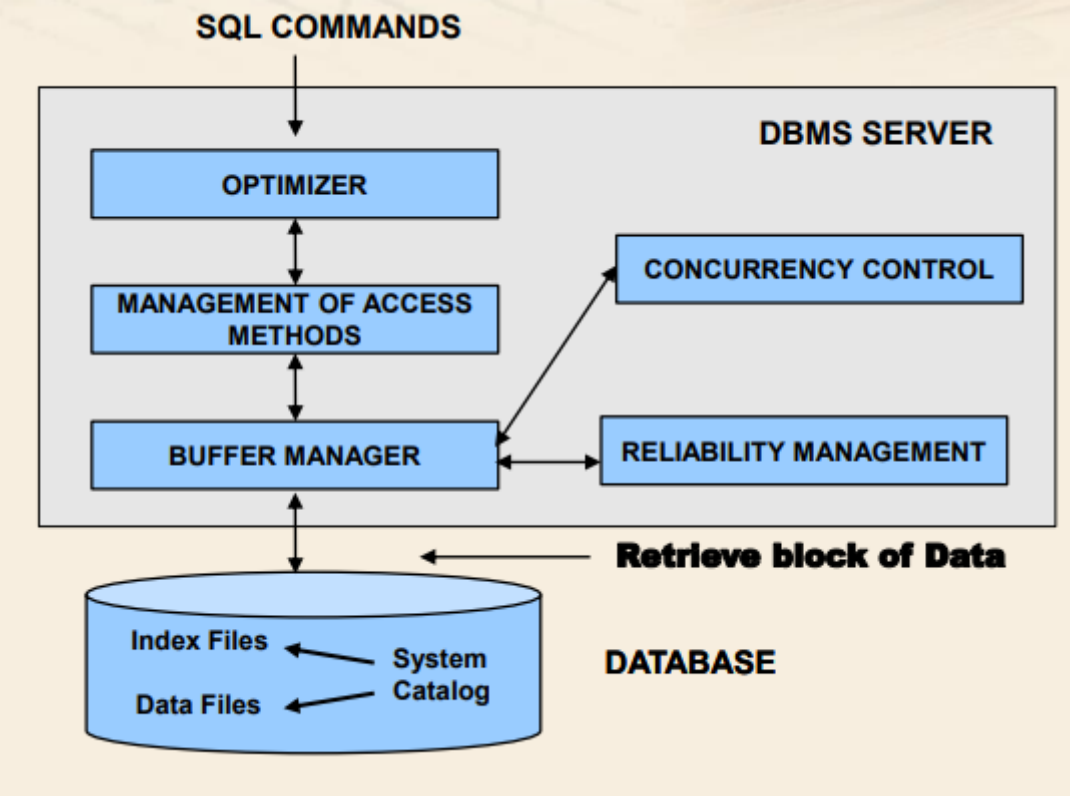
# 7. DBMS Intro

Stands for **D**ata**B**ase **M**anagement **S**ystem and essentially is a software that manages databases.
The following image depicts a simplified architecture of the DBMS:




We can see that it's made of several components:

- **Optimizer:** optimize an input query in the sense that:
  - selects the proper way to access data
  - selects the proper execution plan
  - executes lexical,syntatic and semantic parsing for error detection
  - converts the input query into an internal representation(based on relational algebra)
  - guarantees **data indipendece**, that is, a mapping between a logical descriptio of data to a physical representation of it
- **Access Method Manager:** Using the strategy produced by the **optimizer**, the Access Method Manager:
  - performs *physical* access to data
  - implements the selected *strategy* by the optimizer
- **Buffer Manager:** It manages:
  - *page transfer* between *main memory* and *secondary memory*
  - the *main memory portion* pre allocated for the DBMS

- **Concurrency control:** It manages concurrent access to data
- **Reliability Manager:** guarantees the correctness of the database content in case the system crashes. It also guarantees the atomic execution of transactions.
  It does so by exploiting log files to recover a correct database state.

The optimizer along the access method manager and buffer manager are the modules responsible for the query execution.
The optimizer also is the one performing the hard work.

## Transactions

A logical unit of work performed by an application.
It's a sequence of one or more SQL instructions.
They abide to the ACID properties:

- **A - Atomicity:** a transaction must be either fully executed or not. No intermediary states
- **C - Consistency:** a transaction must not violate any integrity constraints on the database
- **I - Independent:** the execution of a transaction is indipendent from another transaction execution. Enforced by the Concurrency Control unit described above
- **D - Durability:** once a transaction is commited, its effects persist even after a system failures