Introduction

- DBMS = a (complex) software system used by a group of users — each with diff. information need — to:
  (0) Define the structure of the data
  (1) manage (insert, delete, update) the data
  (2) query the data.

- Efficiency is an important requirement.
- Maintain consistency (concurrency control) of data.

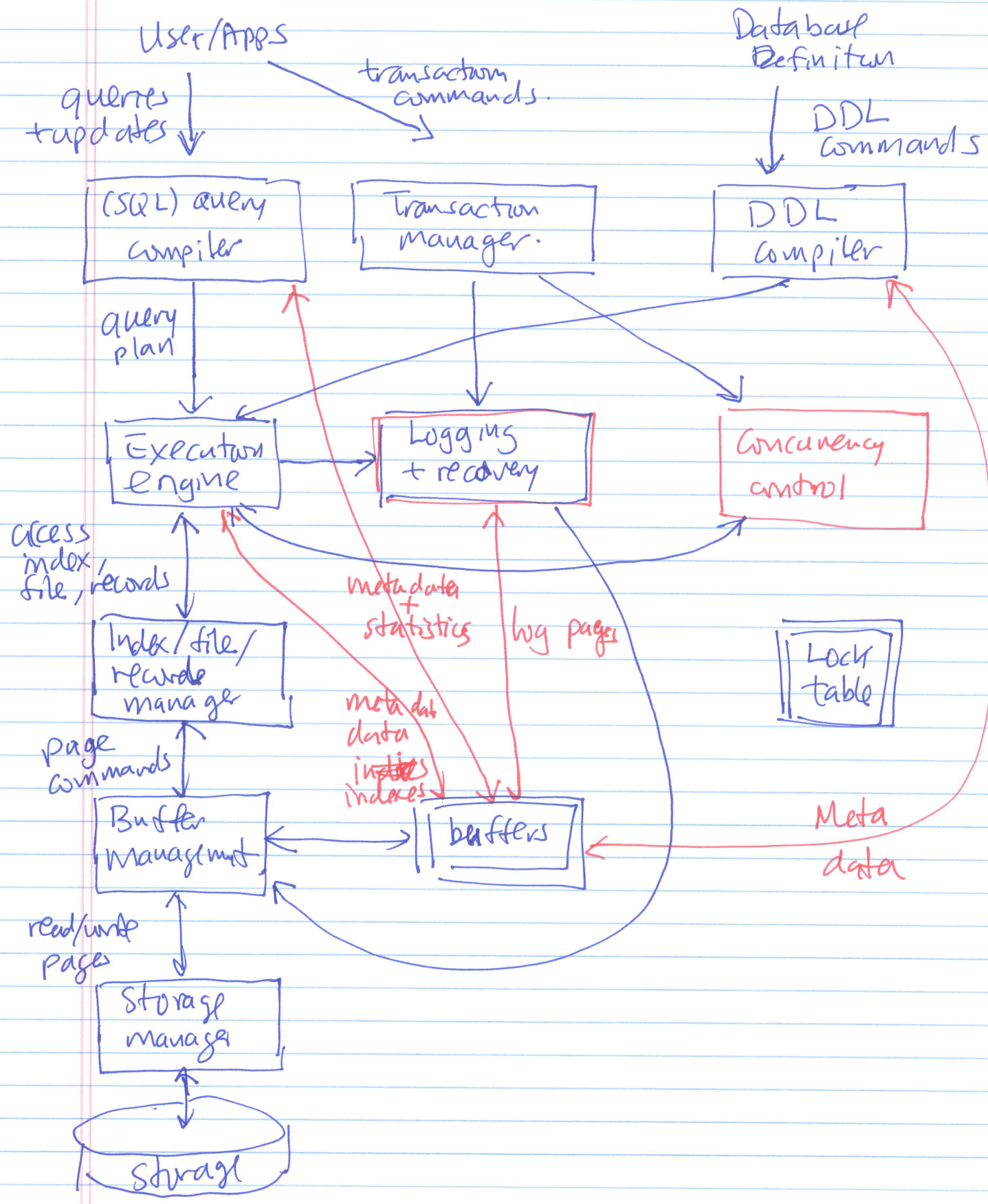
- 2 sources of commands to the DBMS:

  (1) DBA (defining) structure (schema) of the Database (managed by the DBMS)
  (Define Data)

  (2) [User issuing commands interactively
  + Application Programs (no user control).

  (Use of Data) — insert delete update + query.

Overview DBMS



- Defining Data: Data Definition Language (DDL)

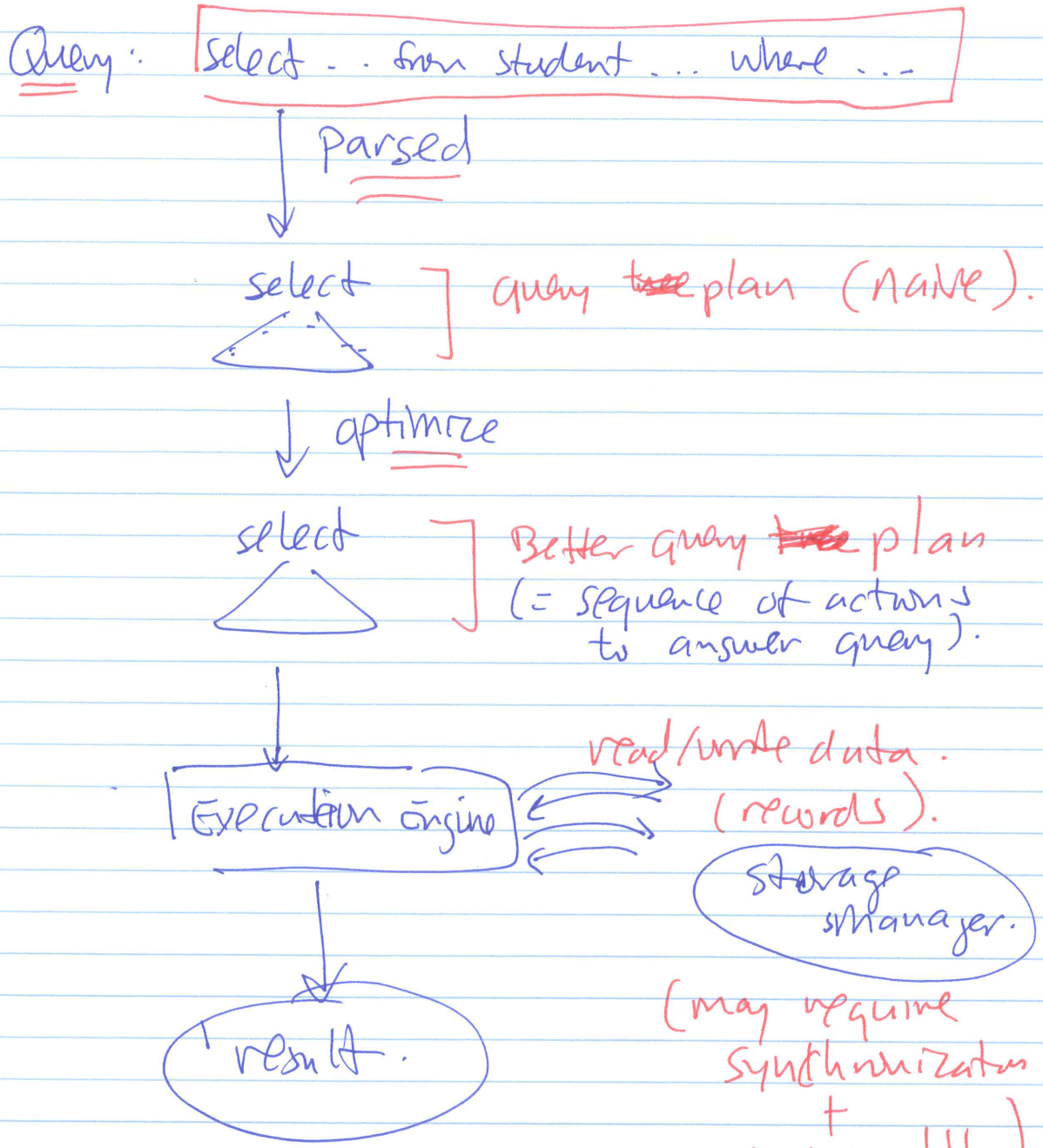
SQL DDL:

```
create table student
  ( studID  char(4) not null,
    name    char(30) not null,
    :
  );
```

- Data Definitions are stored in Database "Catelog" (Meta Data file)

Table Name	Attr. Name	Attr Type	Size
student	studID	char	10
student	name	char	3
:	:	:	:

Overview Query Processing



Concurrency Control

- Transactions must appear to execute in isolation (see no effect of other transactions)
- Eg: Concurrent execution of transactions

T_1: $x = x+1$ $\quad$ T_2: $x = x-1$.

read(x)	
$x = 4$	read(x)
	$x = 4$
$x = x+1$	
$x = 5$	$x = x-1$
	$x = 3$
write(x)	
$x = 5$	write(x)
	$x = 3$

$x = 4$

T_1 T_2 or T_2 T_1

$x = 4$.

But if not isolated → $x = 3$ or $x = 5$ can be the answer!!

- Common Solution: Use locks.

problem: dead lock. (later)

Overview Transaction Processing:

Transaction: a group of operations that must be executed atomically and the effect of the execution must be:

(1) persistent. (durable). even if the system fails

(2) consistent even when multiple transactions are process simultaneously.

Transaction Processing: (processes)

- Concurrency Control manager (scheduler).
  - make sure
    - (1) transactions are atomic
    - (2) transactions are isolated from each other.
- logging + recovery manager
  - ensure persistency (durability) of committed transactions.

Logging: ensure **durability**
(in the face of system failure)

Eg: $\boxed{x=4}$

T_1: $x = x+1$.

read(x)

$x = 4$.

$x = x+1$

write(x)

$\boxed{x=5}$. ← still in memory buffer.

transaction (update) was committed. → commit transaction.

← fails

write(x)

buffer flushed later.

$x=5$ not updated.

use logs.

☆ (may be used by other transactions)

Types of Data used by the DBMS

(1) User Data (actual content of the Database)

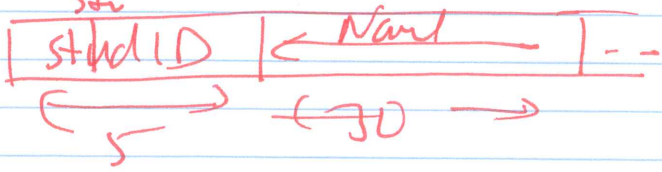
(2) Meta Data (Data that describes the file structures of the user Data.

Eg:

Student	StudID	char	5
Student	Name	char	30

⋮

Student File Record



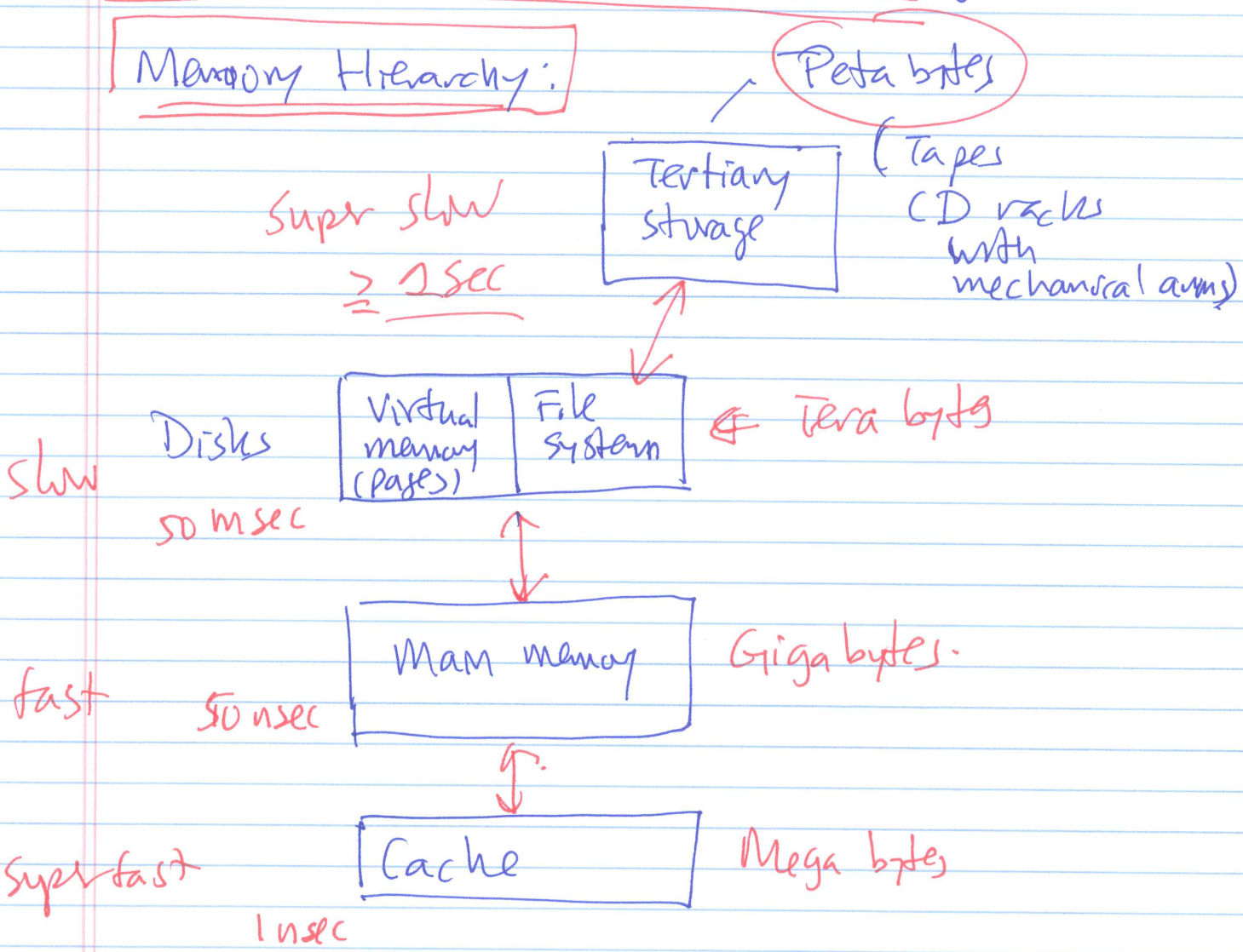
(3) Statistics on (some) data files.

- # records.
- Max value of some field
- Avg value etc.

(4) Indexes: data structures (stored in files) that support efficient access to user data.

Main-Memory Buffers and Buffer Manager.

Memory Hierarchy:



Facts: Data can ONLY be processed when it resides in

Main Memory.

Size is SMALL relative to the amount of data store in secondary / Tertiary storage.

Buffer Manager: Software system responsible for:

(1) Partition the available main memory into buffers (fixed size).

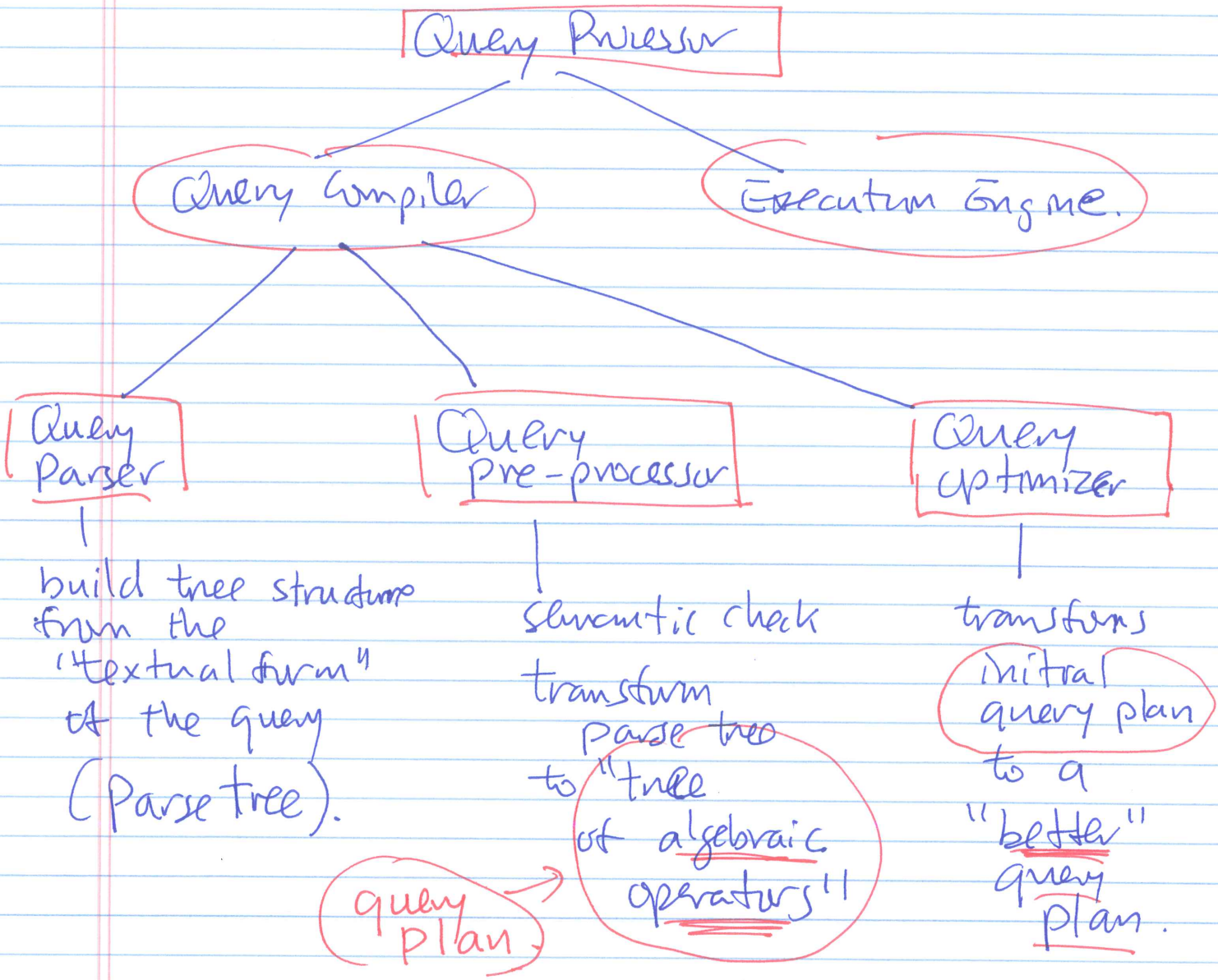
(2) Manage the buffers so that:

data needed to process query is available as soon/frequent as possible in Main Memory

Query + Query Processing:

- Query = asking the DBMS to retrieve information that satisfy certain criteria (boolean condition)
  - condition.

- Query Processor:



Execution Engine

executes queries and other transactions concurrently.

will use — buffer manager
— transaction manager
— concurrency manager.

Course Overview

(1) Storage management
- Disk structure
- File structures.
- Single Dim. Index (B-tree)
- Multi-Dim. Index.

(2) Query Processing.
- Basic query execution.
- Query compiler
- Query optimization

(3) Transaction Processing.
- Concurrency control. (locks, timestamps).
- Logging, recovery