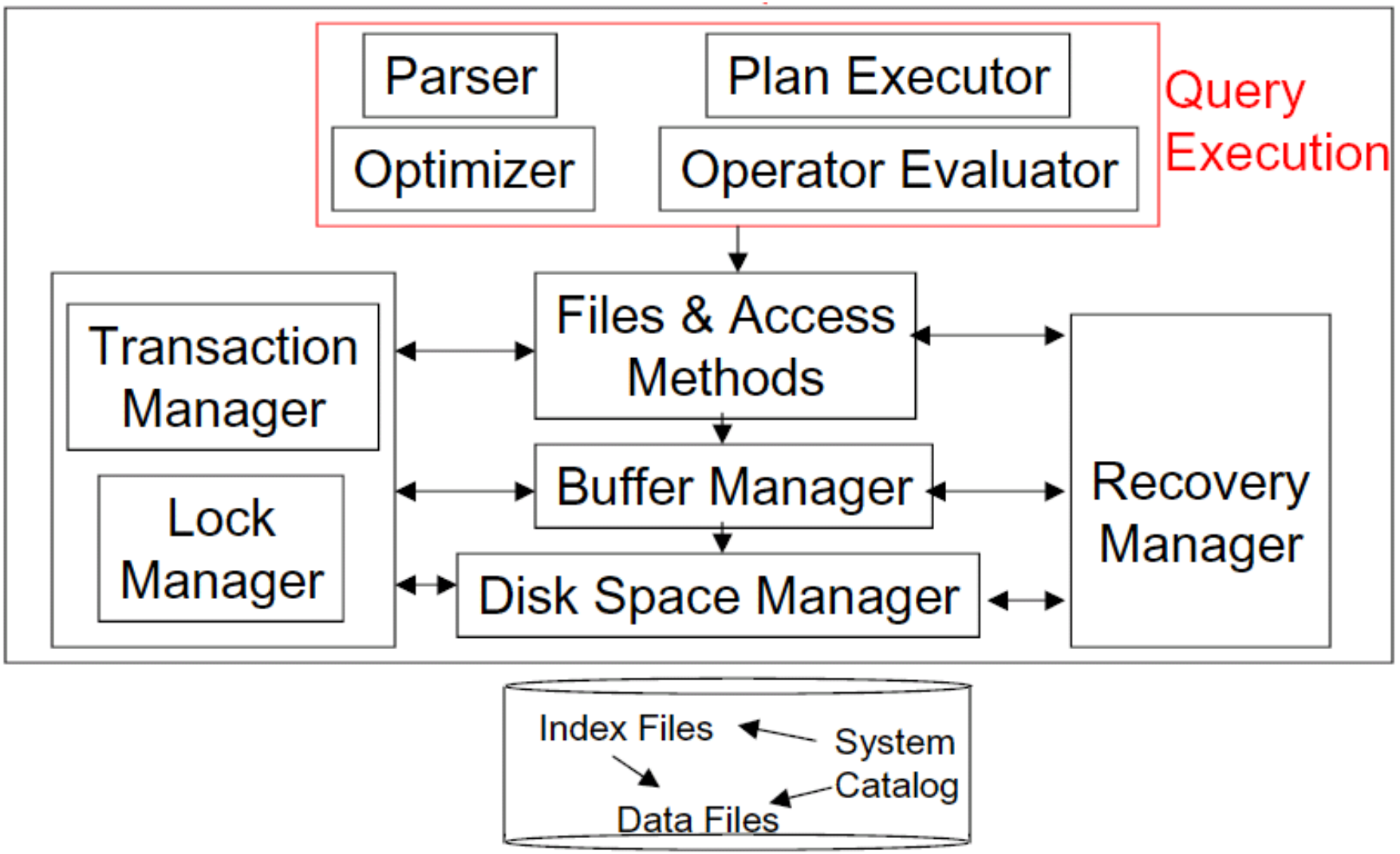
Sisteme de Gestiune
a
Bazelor de Date

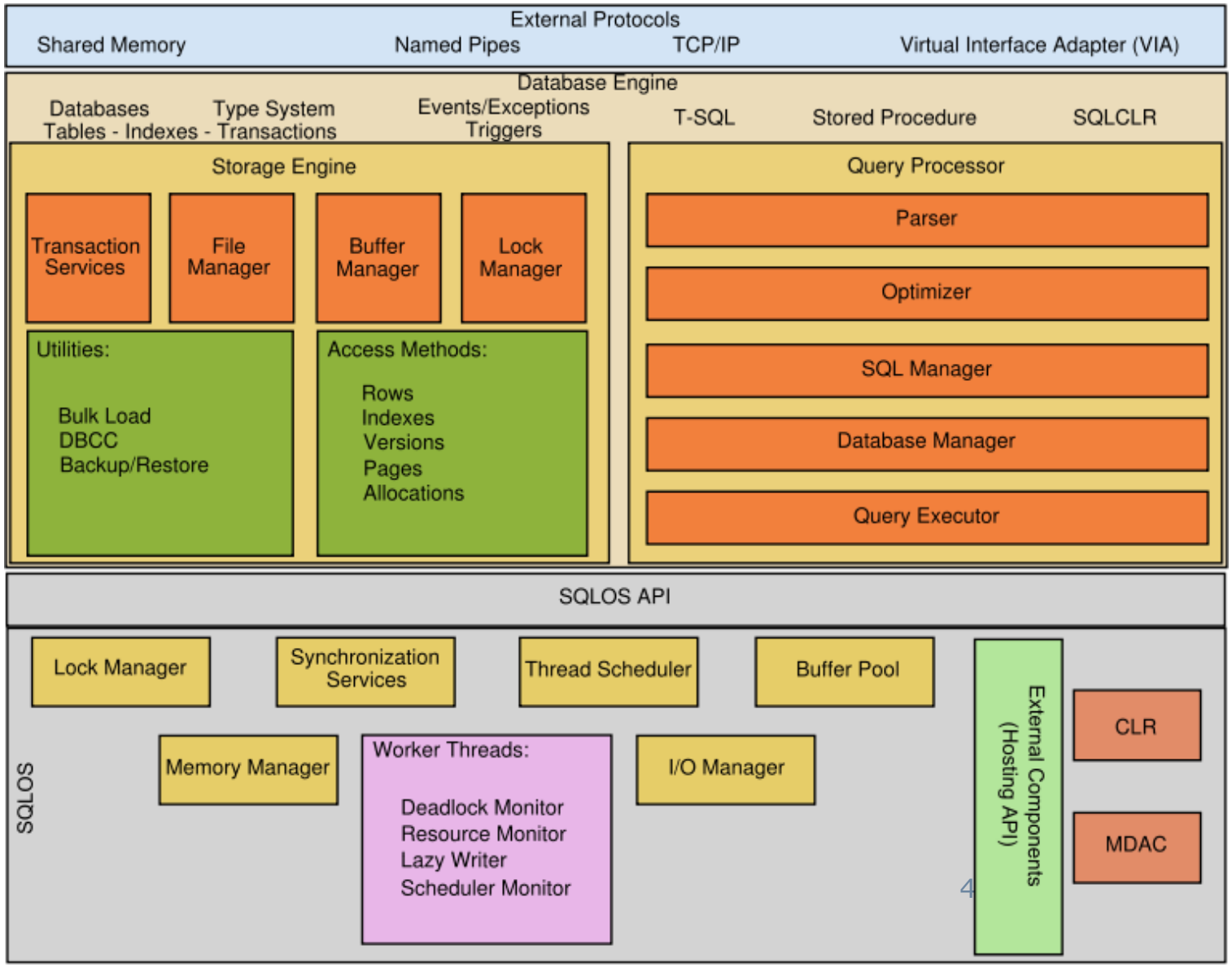
Conținut curs

- Gestiunea tranzacțiilor
- Controlul concurenței
- Recuperarea datelor
- Sortare externă
- Evaluarea operatorilor relaționali
- Optimizarea interogărilor
- Baze de date distribuite / paralele
- Securitate

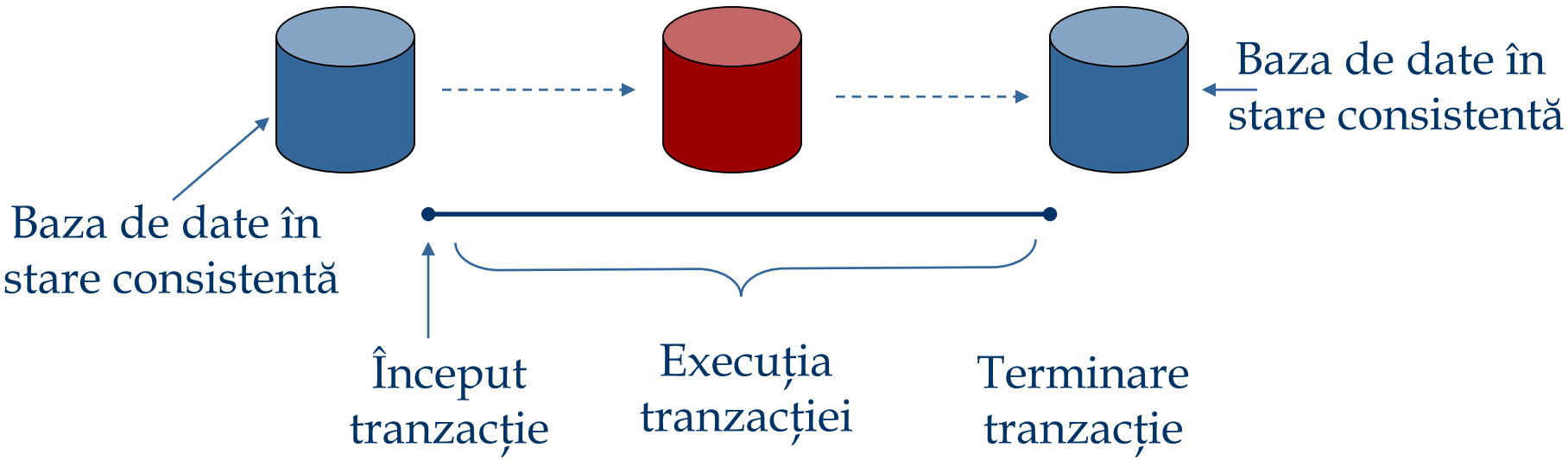
Structura unui SGBD



Structura MS SQL Server



Tranzacții


Baza de date în stare consistentă
Baza de date în stare consistentă
Început tranzacție
Execuția tranzacției
Terminare tranzacție

Tranzacții (cont.)

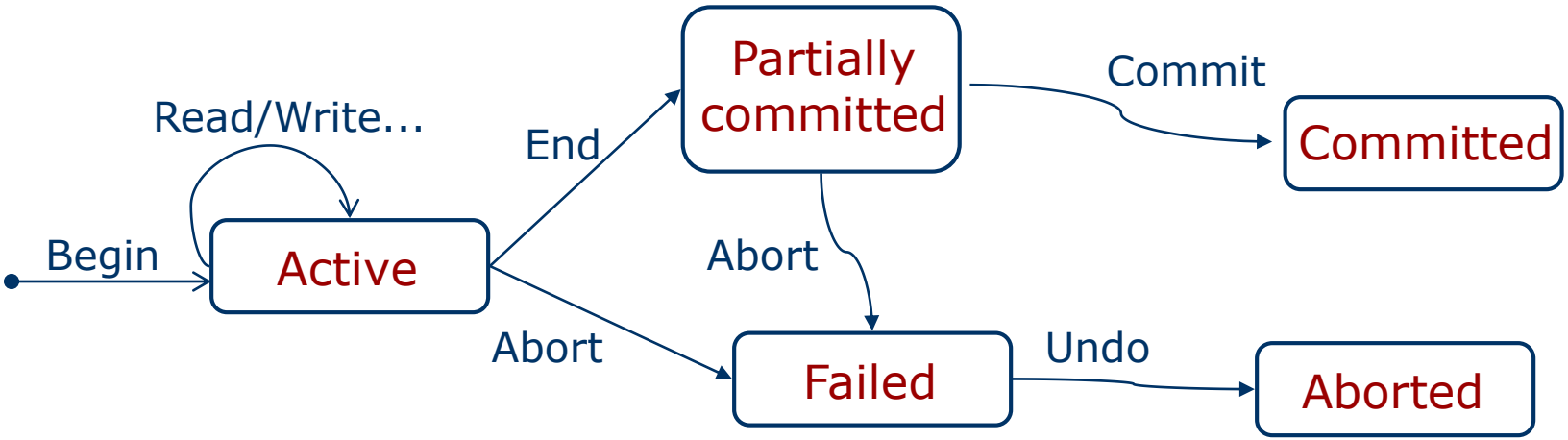
- Execuția concurentă este esențială pentru performanța unui SGBD
  - Deoarece harddisk-ul este accesat frecvent, iar accesul este relativ lent, este preferabil ca CPU-ul să fie "ocupat" cu alte task-uri executate concurent.

- SGBD-ul "vede" un program ce interacționează cu baza de date ca o secvență de operații de **citire** şi **scriere**.

- Begin – transaction
- Read
- Write
- End – transaction
- Commit – transaction
- Abort – transaction
- Undo
- Redo

Stările tranzacțiilor

- Active: tranzacția este în execuție
- Partially Committed: tranzacția urmează să se finalizeze
- Committed: terminare cu succes
- Failed: execuția normală a tranzacției nu mai poate continua
- Aborted: terminare cu *roll back*



Concurența într-un SGBD

- Un utilizator transmite unui SGBD mai multe tranzacții spre execuție:
  - Concurența este implimentată de SGBD prin intercalarea operațiilor mai multor tranzacții (citiri/modificări ale obiectelor bazei de date)
  - Fiecare tranzacție trebuie să lase baza de date într-o stare consistentă
    - Constrângeri de integritate (intră in responsabilitatea SGBD).
    - SGBD nu "înțelege" semantica datelor (intră în responsabilitatea programatorului).

- *Probleme*: Efectul *intercalării* tranzacțiilor şi *blocări*.

Proprietățile tranzacțiilor - **ACID**

- **A**tomicitate (*totul sau nimic*)
- **C**onsistență (*garantare constrângeri de integritate*)
- **I**zolare (*concurența este invizibilă* $\rightarrow$ *serializabilitate*)
- **D**urabilitate (*acțiunile tranzacțiilor executate persistă*)

Atomicitate

- O tranzacție se poate termina cu succes, după execuția tuturor acțiunilor sale, sau poate eşua (uneori forțat de SGBD) după execuția anumitor acțiuni.
- Utilizatorii (programatorii) pot privi o tranzacție ca o operație indivizibilă.
  - SGBD salvează in *loguri* toate acțiunile unei tranzacții pentru a le putea anula la nevoie.
- Acțiunea prin care se asigură atomicitatea tranzacțiilor la apariția unor erori poartă numele de recuperarea datelor (*crash recovery*)

Consistență

- O tranzacție executată *singură* pe o bază de date consistentă, lasă baza de date într-o stare consistentă.
- Tranzacțiile păstrează constrângerile de integritate ale bazelor de date.
- Tranzacțiile sunt programe *corecte*

Isolation

- Dacă mai multe tranzacții sunt executate concurent, rezultatul trebuie sa fie identic cu una dintre execuțiile seriale a acestora (indiferent de ordine) - *serializabilitate*.
- O tranzacție nu poate partaja modificările operate până nu este finalizată
  - Condiție necesară pentru evitarea eşecurilor in cascadă.

Durabilitate

- Odată o tranzacție finalizată, sistemul trebuie să garanteze că rezultatul operațiilor acesteia nu se vor pierde, chiar si la apariția unor erori sau blocări ulterioare.

- Recuperarea datelor

Exemplu

T1: BEGIN A=A+100, B=B-100 END

T2: BEGIN A=1.06*A, B=1.06*B END

- Prima tranzacție transferă 100€ din contul B în contul A.
- Cea de-a doua tranzacție adaugă o dobandă de 6% sumelor din ambele conturi.

Exemplu

- O posibilă intercalare a operațiilor (*plan*):

T1: A=A+100, B=B-100
T2: A=1.06*A, B=1.06*B

- O a doua variantă:

T1: A=A+100, B=B-100
T2: A=1.06*A, B=1.06*B

Cum "vede" SGBD al doilea plan:

T1: R(A), W(A), R(B), W(B)
T2: R(A), W(A), R(B), W(B)

Operații conflictuale

- Daca două tranzacții citesc date, acestea nu intră in conflict şi ordinea de execuție nu este importantă
- Dacă două tranzacții citesc şi modifică date complet separate, acestea nu intră in conflict şi ordinea de execuție nu contează.
- Dacă o tranzacție modifică o dată iar o altă tranzacție citeşte sau modifică aceeaşi dată, ordinea de execuție este importantă.
  - Conflict WR : T2 citeşte date modificate anterior de T1
  - Conflict RW: T2 modifică date citite anterior de T1
  - Conflict WW: T2 modifică date modificate anterior de T1

Anomalii ale execuției concurente

- *Reading Uncommitted Data* (conflict WR, "*dirty reads*"):

T1: R(A), W(A), R(B), W(B), **A**
T2: R(A), W(A), **C**

- *Unrepeatable Reads* (conflict RW):

T1: R(A), R(A), W(A), **C**
T2: R(A), W(A), **C**

- *Overwriting Uncommitted Data* (Conflict WW, "*blind writes*"):

T1: W(A), W(B), **C**
T2: W(A), W(B), **C**

- Nota finală
  - 50% - activitate laborator / test practic
  - 50% - examen scris (*în timpul semestrului!*)