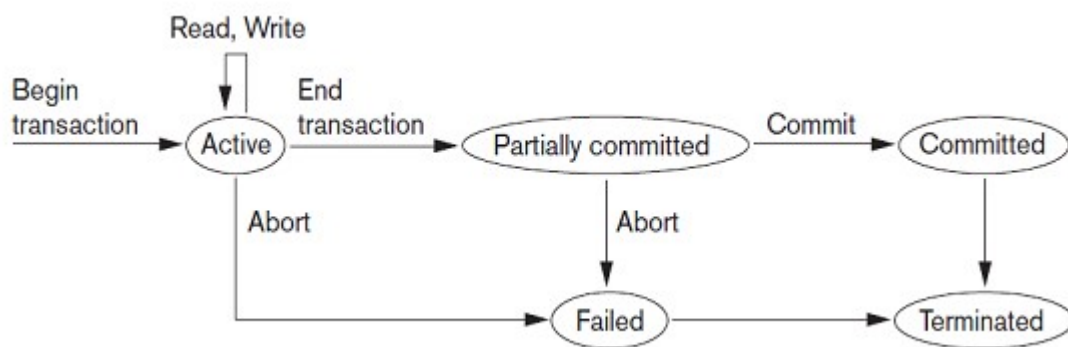


- **Exercises from chapter 08 Introduction to Transactions**

- **Questions**

- 1. Analyze the ACID properties of transactions.
  - Atomicity: The transaction is executed entirely. Or is executed or not.
  - Consistency preservation: The database will always be maintained in a consistent state, complying with integrity restrictions.
  - Isolation: property to make a lot of transactions at a same time in a database without interferences with other transactions.
  - Durability or permanency: Ensures that modifications made to the database are stored permanently.
- 2. Draw the state diagram of a transactional and analyze each of them.

Which is the difference between partially committed and committed?



When the transaction is ended, is partially committed (the transaction is confirmed by the user and if abort the transaction comes to failed). After that when the DBMS confirm the transaction this is committed 100% and the transaction comes to terminated.

3. What is understood by concurrent execution of transactions in databases in a multi-user environment?

Analyze why concurrency control is needed and provide some exemples.

- A lot of user can do queries in a database at a same time like no one is doing with the isolation propertie.

In a ticket sale of a big event. In a fly booking system. This is necessary to garantee that more than one person cant buy the same place.

4. What is the log file used for?

- Is the file that stores all the movements made in the database sequentially and is not altered by basic system failures.

5. Explain what problems can arise when there is concurrent access to data and explain each one.

- Lost update: When different transactions use and modify the same values, without taking into account the possible modifications that may have been made to each of them throughout the execution.
- Temporary update problem: If a transaction reads a value updated from another transaction and this transaction fails after.
- Incorrect summary problem: When we made a summary operation and one of the values is updated before the transaction is finished.
- Unrepeatable read problem: If a transaction read the same value twice, can take different values for the same read.

6. Relate each section with the corresponding concept (one of them is related to two).

The system log	- Keeps track of all operations carried out by transactions.
----------------	--

single-user DBMS	- At most a single user can use the system at a time
Multiprogramming	- Allows the operating system to run multiple processes concurrently
Granularity	- Size of the elements stored in the database
Transaction	- It forms a logical unit of processing
DBMS multi-user	- Systems with large databases and hundreds of concurrent users - Several users can access the database concurrently
Begin and end states of a transaction	- Specify the transaction boundaries
Transaction Processing systems	- Require high availability and fast response time

7. Given the following transaction plans, indicate which operations are in conflict and what operations are not (give two examples). Justify your answer.

Operation	abbreviation
Read element	r
Write element	w
Commit	c
Rollback	r

P1= r1(x); r2(x); w1(x); r1(x); w2(x); c2; w1(y); c1;

In conflict:

- r1(x) and w2(x)
- r2(x) and w1(x)

Not in conflict:

- $r_1(x)$  and  $r_2(x)$  both are read operation
- $r_1(x)$  and  $r_1(x)$  are the same operation

$P_2 = r_1(x); w_1(x); r_2(x); w_2(x); c_2; r_1(y); c_1;$

In conflict:

- $r_1(x)$  and  $w_2(x)$
- $w_1(x)$  and  $r_2(x)$
- $w_1(x)$  and  $w_2(x)$

Not in conflict:

- $r_1(x)$  and  $w_1(x)$  are same transaction
- $r_2(x)$  and  $w_2(x)$  are same transaction