DBMS

PANA ACADEMY

Relation Algebra

Relational Algebra is a procedural query language.

As it is pure mathematics, there is no use of English Keywords in Relational Algebra and operators are represented using symbols.

Symbols used in Relational Algebra

These are the basic/fundamental operators used in Relational Algebra.

1. Selection(σ)
2. Projection(π)
3. Union(U)
4. Set Difference(-)
5. Set Intersection(∩)
6. Rename(ρ)
7. Cartesian Product(X)

1. Selection(σ): It is used to select required tuples of the relations.

σ(c>3)R will select the tuples which have c more than 3.

2. Projection(π): It is used to project required column data from a relation.

```
π(Student_Name)FRENCH U π(Student_Name)GERMAN
```

3. π(Student_Name)FRENCH - π(Student_Name)GERMAN is for set difference.

Relational Algebra Symbols

Rename(ρ): Rename is a unary operation used for renaming attributes of a relation.

`ρ(a/b)R` will rename the attribute 'b' of the relation by 'a'.

Estimating Query Cost

- Number of disk accesses
- Execution time taken by the CPU to execute a query
- Communication costs in distributed or parallel database systems.

Query Processing Steps

1. Parsing and translation
2. Optimization
3. Evaluation

select emp_name from Employee where salary>10000;

Parser will convert sql statement into relational algebra which looks like

- $\pi_{salary}\ (\sigma_{salary>10000}\ (Employee))$

Thus, after translating the user query, the system executes a query evaluation plan.

Query Optimization

Usually, a database system generates an efficient query evaluation plan, which minimizes its cost. This type of task performed by the database system and is known as Query Optimization.

Finally, after selecting an evaluation plan, the system evaluates the query and produces the output of the query.

Transaction

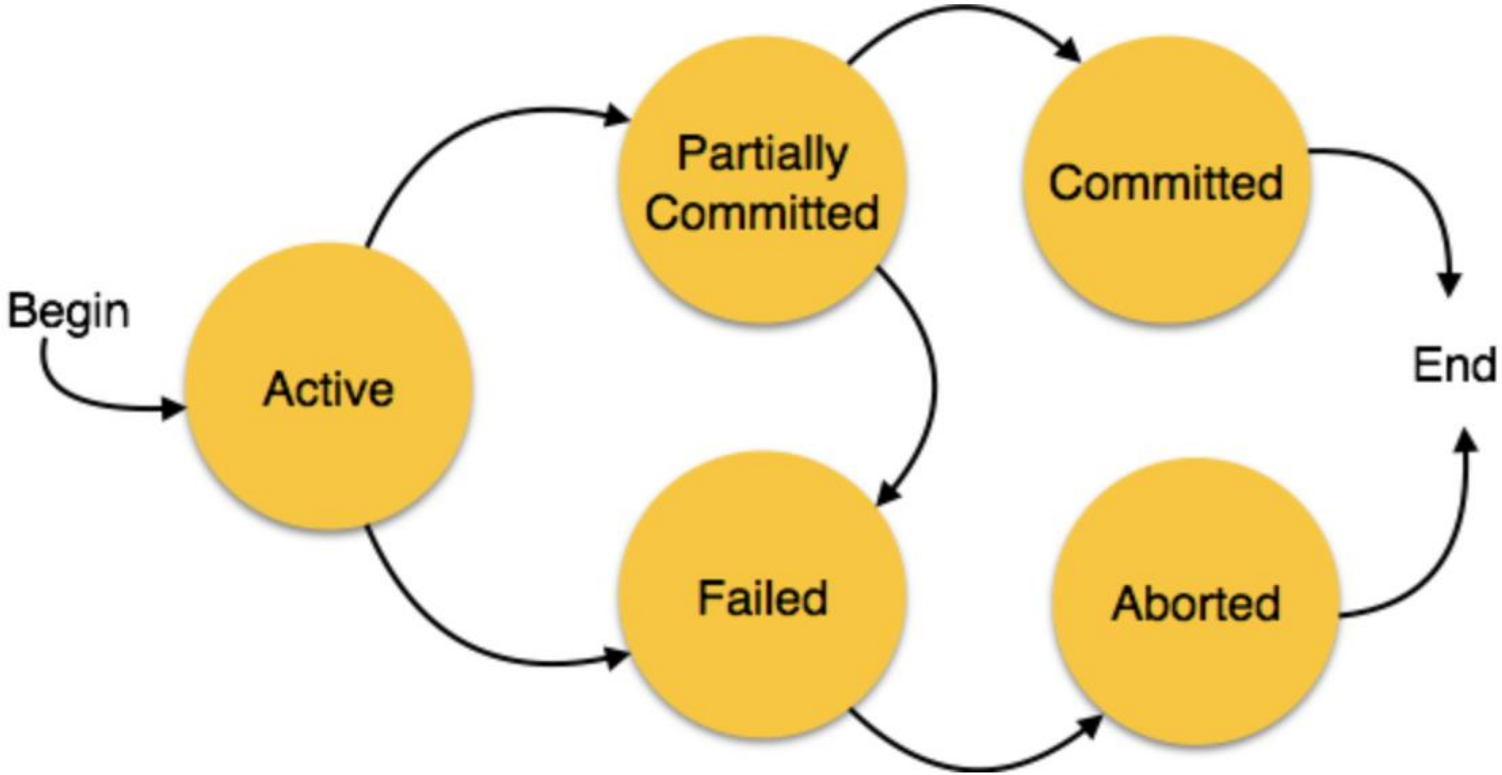
Transaction in Database Management Systems (DBMS) can be defined as a set of logically related operations.

A transaction can be defined as a group of tasks.

Let’s take an example of a simple transaction. Suppose a bank employee transfers Rs 500 from A's account to B's account. This very simple and small transaction involves several low-level tasks.

States of Transactions

A transaction in a database can be in one of the following states −



States of Transaction

Active – In this state, the transaction is being executed. This is the initial state of every transaction.

Partially Committed – When a transaction executes its final operation, it is said to be in a partially committed state.

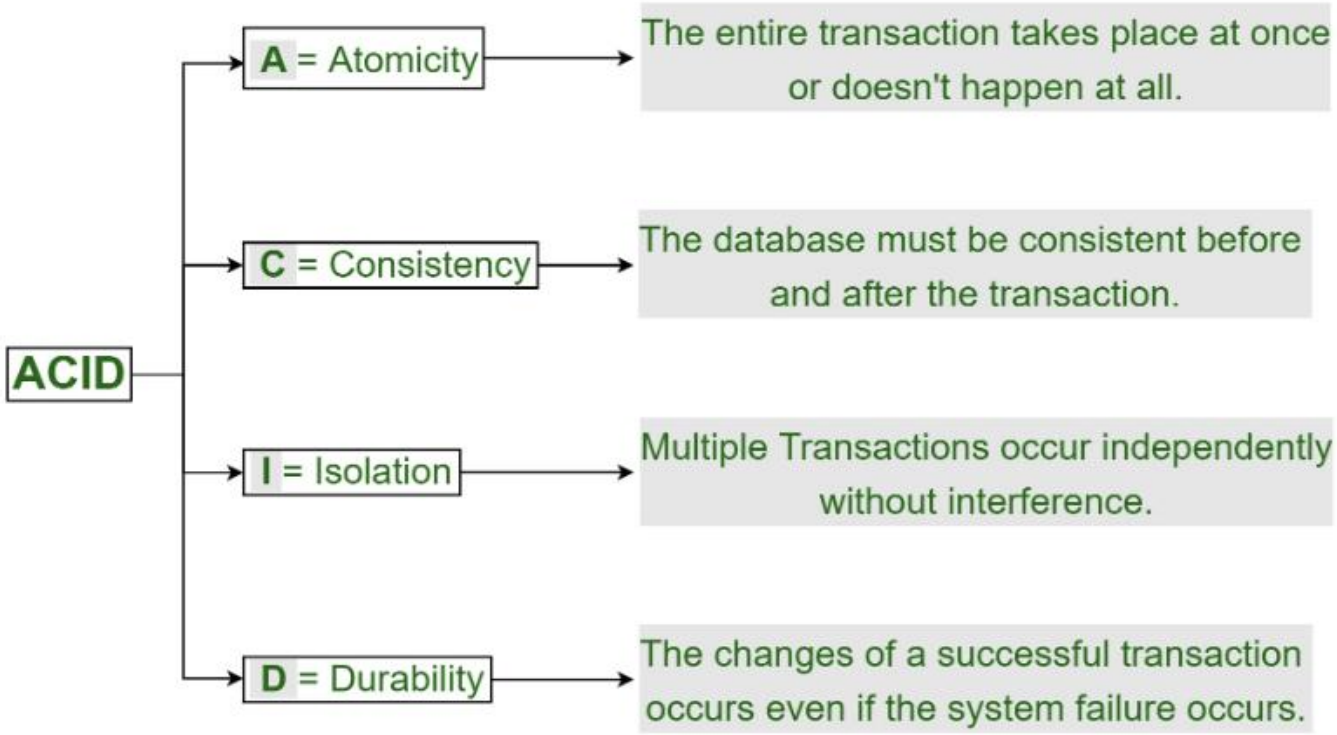
Failed – A transaction is said to be in a failed state if any of the checks made by the database recovery system fails. A failed transaction can no longer proceed further.

Aborted – If any of the checks fails and the transaction has reached a failed state, then the recovery manager rolls back all its write operations on the database to bring the database back to its original state where it was prior to the execution of the transaction. Transactions in this state are called aborted. The database recovery module can select one of the two operations after a transaction aborts –

- Re-start the transaction
- Kill the transaction

Committed – If a transaction executes all its operations successfully, it is said to be committed. All its effects are now permanently established on the database system.

ACID Properties in DBMS



Atomicity

By this, we mean that either the entire transaction takes place at once or doesn't happen at all. There is no midway i.e. transactions do not occur partially. Each transaction is considered as one unit and either runs to completion or is not executed at all. It involves the following two operations.

— **Abort** : If a transaction aborts, changes made to the database are not visible.

— **Commit** : If a transaction commits, changes made are visible.

Atomicity is also known as the 'All or nothing rule.

Consistency

This means that integrity constraints must be maintained so that the database is consistent before and after the transaction.

Isolation

Changes occurring in a particular transaction will not be visible to any other transaction until that particular change in that transaction is written to memory or has been committed. This property ensures that the execution of transactions concurrently will result in a state that is equivalent to a state achieved these were executed serially in some order.

Durability:

This property ensures that once the transaction has completed execution, the updates and modifications to the database are stored in and written to disk and they persist even if a system failure occurs. These updates now become permanent and are stored in non-volatile memory. The effects of the transaction, thus, are never lost.

Example of a case

T	T''
Read (A) A: = A * 100 Write (A) Read (B) B: = B - 50 Write (B)	Read (A) Read (B) Z: = A + B Write (Z)

Case example

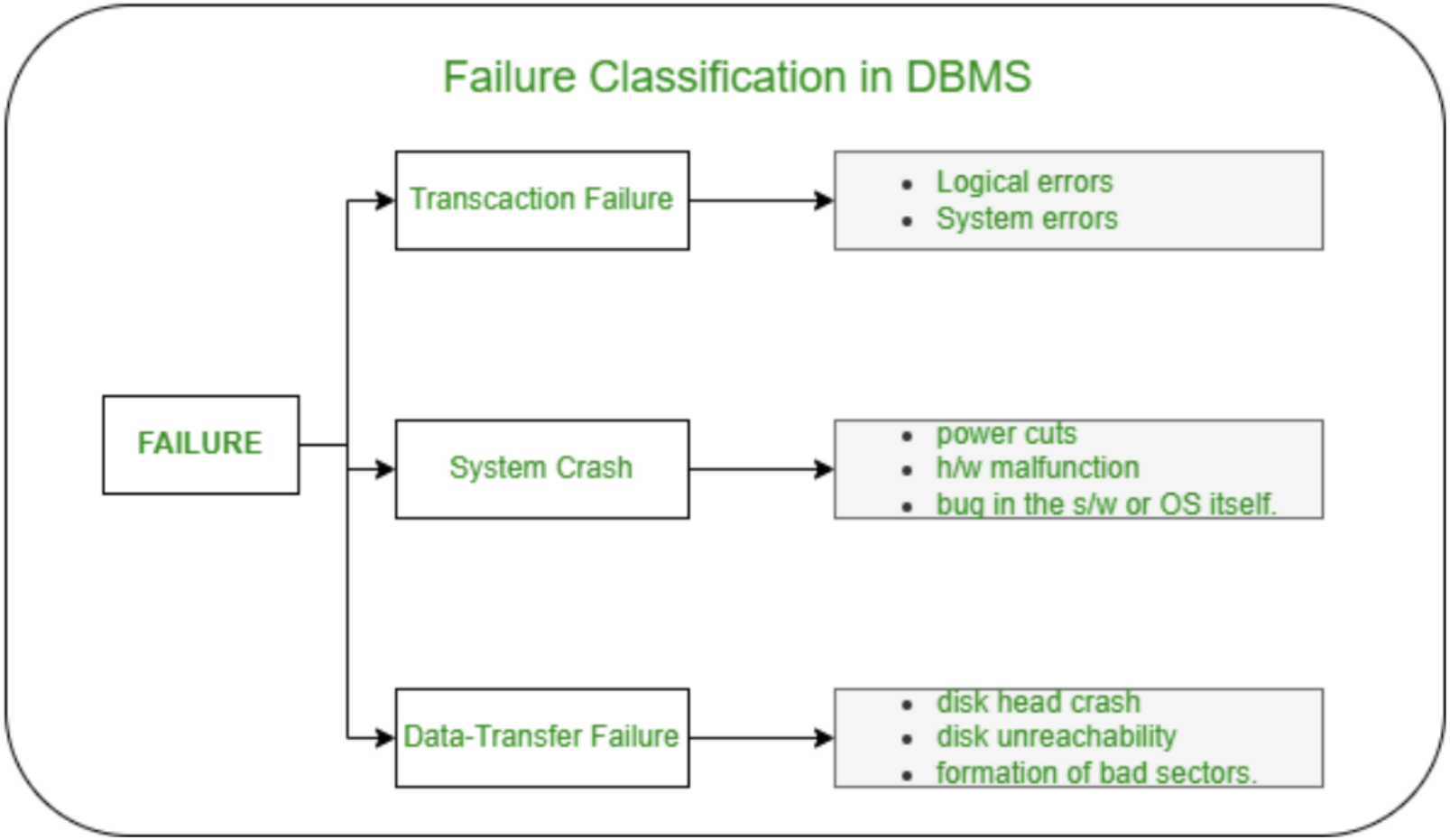
Suppose that T has been executed here till Read(B) and then T'' starts. As a result, the interleaving of operations would take place. And due to this, T'' reads the correct value of A but incorrect value of B.

T'': (X+B = 50, 000+500=50, 500)

Thus, the sum computed here is not consistent with the sum that is obtained at the end of the transaction:

T: (A+B = 50, 000 + 450 = 50, 450).

It results in the inconsistency of a database due to the loss of a total of 50 units. The transactions must, thus, take place in isolation. Also, the changes must only be visible after we have made them on the main memory.



Failure Classification in DBMS

Mcq practice

https://www.sanfoundry.com/rdbms-questions-answers-transaction-concept-model/

https://www.sanfoundry.com/database-mcqs-relational-algebra/

https://www.sanfoundry.com/rdbms-multiple-choice-questions-answers/