**Database Transaction and Recovery**

**Transactions –** a sequence of database operations such as updating or deleting. They either complete or fail (all or nothing) – this ensures that if transaction isn’t complete in its entirety, then the database isn’t impacted. Only fully completed transactions are committed to the database.

**Transaction Properties – ACID State**

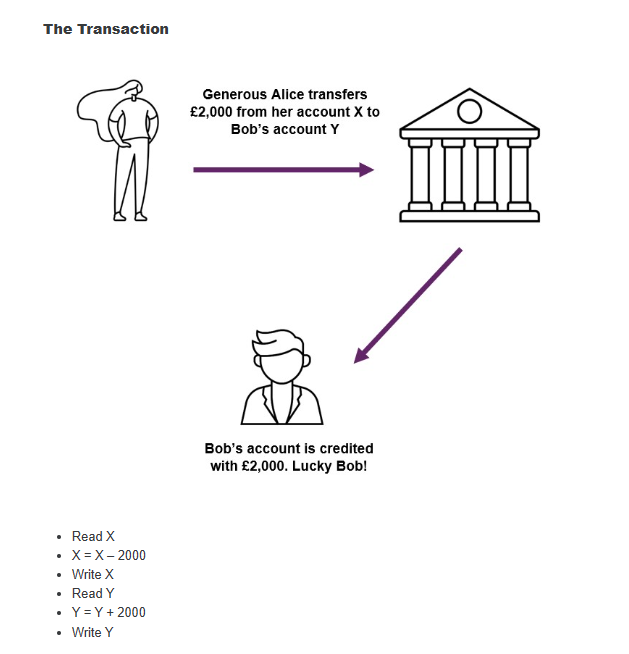
Source: ACID properties of transactions - <https://www.ibm.com/docs/en/cics-ts/5.4?topic=processing-acid-properties-transactions>

**Atomicity -** All changes to data are performed as if they are a single operation. That is, all the changes are performed, or none of them are.

**Consistency -** Data is in a consistent state when a transaction starts and when it ends.

**Isolation -** The intermediate state of a transaction is invisible to other transactions. As a result, transactions that run concurrently appear to be serialized.

**Durability -** After a transaction successfully completes, changes to data persist and are not undone, even in the event of a system failure.

In example below, either money is added to Bob’s account and deducted from Alice’ account or neither action is performed!  


Atomicity - either everything in the transaction executes or nothing at all.

Consistency - Data is in a consistent state when a transaction starts and when it ends. A transaction transforms the database from one consistent state into another consistent state.

Isolation - Transactions are isolated from one another. Changes made by a transaction are not visible to other transactions until it finishes (committed/rolled back). Isolation prevents Dirty/Nonrepeatable reads.

Durability - The effect of completed transactions are persistent despite failures. Committed changes are irreversible, even in the event of a system failure.

**Interleaving –** Makes a system more reliable and efficient by arranging data in a noncontinuous manner. It divides data into chunks so that smaller sections can be accessed more quickly via increased bandwidth. Processer can fetch and send more data to and from memory in same time as it’s divided up. Leads to more efficient systems

Source: Techopedia -What Does Interleaving Mean? <https://www.techopedia.com/definition/5683/interleaving>

**The Transaction Manager –** Enforces the ACID process by scheduling the operation of transactions. IT keep track of all operations, logging them so on failure the most recent can be accessed as the reset position.

Transactions committed to database when a commit request occurs. Transaction manger informs whether the process completed or aborted.

**The Use of Commit and Rollback**

Successful transactions are committed, unsuccessful will roll back. Only successful are visible. ACID principle

**Two Phase Locking (2PL)**

Controls concurrent use and transaction processing. Applies locs to block two transactions updating data at one time. Read access likely to be shared. If an item is exclusively locked, no other action can access or change it. Transaction log ensures processes are recorded.

**Transaction Logs**

A transaction log records the details of all transactions. It enables roll back and ensures durability as changes that were not completed before a failure.

Shows all changes to the database and how to undo them, so original and new values and time transactions completed with status.

**Scheduling Transactions**

This is a list of actions from a set of simultaneous transactions. A well-formed will have transactions in order they run. There are different types:

Serial Schedule: No interleaving of different transactions.

Equivalent Schedules: Same result on any database state.

Serialisable Schedule: Non-serial schedule that allows concurrent execution without interference.

**System Failures and Checkpoints**

A system failure affects all currently running transactions. Checkpoints are used to identify how far back into log of transactions we need to search for last good state in terms of transaction processing.

A checkpoint is a point at which synchronisation occurs between the database and the log file. A checkpoint is added to the log file with details of which transactions are currently running.

**System Failures**

To recover all transactions which were running when the failure happened need to be undone and restarted, and all transactions which committed after the last checkpoint need to be redone. Only updates since the last checkpoint are affected when system fails. They can be recovered from the log.

**Media Failures**

These are more serious as data stored on the disk may be damaged including the log. Backups protect against these occurrences, log and entire database written to secondary storage. This is time consuming and can require downtime.

Backups should be frequent enough to minimise loss, but not too cause too much disruption. Nightly is typical.

.