Database Model

- A **database** is a set of named **data items** (e.g., x, y, z).
- A **database system** is a collection of hardware and software that support **operations** to access the database.
  - Read operation and write operation.
- Operations are executed **atomically**.
  - As if operations are executed sequentially and one at a time.

- Database system also supports transaction-specific operations
  - Start, commit, abort
Transaction Model

- A transaction $T_i$ is a set of operations and a (partial order).
  - May end with $a_i$ or $c_i$, but not both.
ACID Properties

• **Atomicity**: A transaction executes completely or not at all.
• **Consistency**: The transaction brings the database from one consistent state to another.
• **Isolation**: Each transaction behaves as if it were operating alone with all resources to itself.
• **Durability**: When a transaction has completed successfully, its operations are guaranteed to survive failures.
Histories and Serializability

• A **history** is an execution of multiple transactions.
  • If all transactions have terminating operations (abort or commit), this is a **complete history**.

• A **serial history** is one in which no two transactions are interleaved.

• We want to allow for transactions to interleave, but still guarantee isolation.
  • Want the effects of a history to be the same as some serial history.
Serializability

- A history is **serializable** if its effects are the same as some serial history.

- Two operations **conflict** if they operate on the same data time and at least one of them is a write.

- Two **histories are equivalent** if they contain the same transactions and they order conflicting operations (of non-aborted transactions) in the same way.
Serializability (2)

- A history is *serializable* if it is equivalent to some serial history.