The ACID Principle

- Atomicity
  - Processing transactions **either completely or not at all**

- Consistency
  - Preserving all integrity conditions when processing transactions

- Isolation
  - Processing transactions "as if there was no other"

- Durability
  - Making the effects of transactions **persistent** (after successful commitment)

Goals of ACID for DB applications:

- **don't care** about success of transaction steps (on fault: undo-recovery by DBS & repetition of transaction by application)
- **don't care** about other users (synchronisation by DBS)
- **don't care** about integrity constraints
  - e.g. financial transaction: logical consistency temporally violated

But:
- **care** about clever definition of transactions
Integrity Violation

- Example: Financial transactions
  - accounts A, B
  - DB state: A=0€, B=10€
  - Logical integrity constraint: \( A+B=10€ \)

- Transaction:
  1. Reduce deposit of B by 5€
  2. Increase deposit of A by 5€

- Logical integrity requires physical integrity
- Other class of logical integrity constraints are checked on each step (e.g. value ranges of variables etc.)

Missing Isolation

- Lost Update

- Dirty Read

Missing Isolation

- Phantom Problem

Paper time

Process 1 is expected to compute the sum of \( x, y, z \) at this time.