1. **Atomicity:**
   * *Simple Explanation:* Think of an atomic transaction like a single, indivisible action.
   * *Example:* If you're transferring money between two bank accounts, either the entire transfer happens successfully, or it doesn't happen at all. There's no in-between state where the money is partially transferred.
2. **Consistency:**
   * *Simple Explanation:* After a transaction, the database should be in a valid and expected state.
   * *Example:* If you're updating the quantity of a product in a database, the consistency principle ensures that the product's quantity is a valid number and adheres to any defined rules (e.g., not a negative value).
3. **Isolation:**
   * *Simple Explanation:* Transactions should be isolated from each other until they're completed.
   * *Example:* If two people are booking tickets online simultaneously, one person's transaction shouldn't interfere with the other. They should each see the availability of tickets as if they were the only ones accessing the system.
4. **Durability:**
   * *Simple Explanation:* Once a transaction is committed, its effects should be permanent, even in the face of system failures.
   * *Example:* If you update your address in an online shopping account and th