In the context of a relational database system, a transaction is essentially a sequence of one or more database operations that are treated as a single, indivisible unit of work. This means that either all of the operations within the transaction are successfully completed, or none of them are.

Here's a breakdown of what that means:

- **Unit of Work:**
    - A transaction represents a logical piece of work that you want the database to perform. It might involve reading data, writing data, updating data, or deleting data.
- **All or Nothing:**
    - The key concept is that transactions are atomic. This means they either succeed entirely, or they fail entirely. If any part of the transaction fails, the database rolls back to its original state, as if the transaction never happened.
- **ACID Properties:**
    - Transactions in relational databases are typically governed by the ACID properties:
        - **Atomicity:** All operations within the transaction are treated as a single unit.
        - **Consistency:** The transaction ensures that the database remains in a valid state.
        - **Isolation:** Transactions occur independently of each other.
        - **Durability:** Once a transaction is completed, the changes are permanent.

In simpler terms, imagine a bank transfer: you want to deduct money from one account and add it to another. This is a transaction. You wouldn't want the money to be deducted without being added, or vice versa. Transactions ensure that both actions happen together, or neither happens.