**1. Transaction là gì?**

**Transaction (giao dịch)** là một **nhóm các thao tác** (query) được gom lại và thực hiện như **một đơn vị công việc duy nhất**.

* Nghĩa là: **hoặc tất cả các thao tác đều thành công, hoặc tất cả đều bị hủy**.
* Không bao giờ có tình trạng thực hiện được nửa chừng (một phần thành công, một phần thất bại).

Ví dụ:  
👉 Bạn chuyển tiền từ A → B:

* Giảm số dư tài khoản A.
* Tăng số dư tài khoản B.

Nếu một trong hai thao tác thất bại (ví dụ giảm ở A xong nhưng chưa tăng ở B), thì phải **rollback** để dữ liệu quay lại như ban đầu.

**2. Transaction dùng để làm gì?**

Transaction giúp:  
✅ **Đảm bảo tính toàn vẹn dữ liệu (Data Integrity)**.  
✅ **Tránh mất mát dữ liệu** khi hệ thống gặp lỗi (mất điện, crash server...).  
✅ **Hỗ trợ nhiều người cùng truy cập dữ liệu an toàn** (concurrency control).  
✅ Tuân thủ nguyên tắc **ACID**:

* **A**tomicity (Tính nguyên tử) – tất cả hoặc không gì cả.
* **C**onsistency (Tính nhất quán).
* **I**solation (Tính độc lập).
* **D**urability (Tính bền vững).

**3. Cách sử dụng Transaction**

Tùy vào ngôn ngữ/framework bạn dùng:

**(1) SQL thuần**

START TRANSACTION;

UPDATE accounts SET balance = balance - 1000 WHERE id = 1;

UPDATE accounts SET balance = balance + 1000 WHERE id = 2;

COMMIT; -- Xác nhận thành công

-- Nếu có lỗi:

ROLLBACK; -- Hủy toàn bộ giao dịch

**(2) Spring Boot (Java)**

Trong **Spring**, bạn dùng annotation @Transactional:

@Service

public class TransferService {

@Transactional

public void transferMoney(Long fromId, Long toId, BigDecimal amount) {

accountRepository.decreaseBalance(fromId, amount);

accountRepository.increaseBalance(toId, amount);

}

}

**4. Khi nào nên dùng Transaction?**

Bạn nên dùng khi:

* Các thao tác liên quan **chặt chẽ** với nhau.
* Dữ liệu cần **đồng bộ, toàn vẹn**.
* Có nhiều query cần thực hiện theo **một logic nghiệp vụ**.

Ví dụ:

* Chuyển tiền ngân hàng.
* Đặt hàng online (giảm stock, tạo order, trừ tiền user).
* Ghi log + cập nhật dữ liệu chính.
* Import nhiều dữ liệu mà phải thành công toàn bộ.