**Apply Lock (Áp dụng khóa)** là cơ chế kiểm soát truy cập đồng thời (concurrency control) trong cơ sở dữ liệu, giúp đảm bảo tính nhất quán và tránh xung đột khi nhiều người dùng hoặc tiến trình truy cập cùng một dữ liệu.

**Lock rất quan trọng trong hệ thống OLTP (Online Transaction Processing) để tránh mất dữ liệu hoặc lỗi race condition khi nhiều giao dịch chạy song song.**

**Các loại Lock trong Database**

| **Loại Lock** | **Chức năng** | **Ví dụ thực tế** |
| --- | --- | --- |
| **Shared Lock (S Lock)** | Cho phép nhiều tiến trình đọc cùng một lúc nhưng không thể ghi (read-only). | Nhiều người dùng có thể xem cùng một báo cáo, nhưng không thể cập nhật dữ liệu. |
| **Exclusive Lock (X Lock)** | Chỉ cho phép một tiến trình đọc và ghi, các tiến trình khác phải chờ. | Khi một người dùng cập nhật thông tin sản phẩm, người khác không thể chỉnh sửa cùng lúc. |
| **Update Lock (U Lock)** | Giống Shared Lock nhưng có thể nâng cấp lên Exclusive Lock nếu cần. | Khi kiểm tra số lượng hàng tồn kho trước khi cập nhật đơn hàng. |
| **Intent Lock (IX, IS, IU)** | Đánh dấu một bảng hoặc hàng có thể bị khóa để tránh xung đột. | Một transaction muốn lock một hàng trong bảng, trước tiên sẽ đặt Intent Lock lên bảng đó. |
| **Table Lock** | Khóa toàn bộ bảng, không cho phép truy vấn hoặc thay đổi. | Khi chạy một báo cáo lớn trên toàn bộ bảng dữ liệu. |
| **Row Lock** | Khóa một dòng cụ thể để các tiến trình khác không thể cập nhật dòng đó. | Khi một khách hàng cập nhật hồ sơ cá nhân của họ. |
| **Deadlock** | Xảy ra khi hai giao dịch chờ nhau giải phóng lock, gây treo hệ thống. | Giao dịch A lock hàng 1, Giao dịch B lock hàng 2, nhưng cả hai lại cần lock hàng còn lại để tiếp tục. |

**Cách Apply Lock trong SQL**

**Shared Lock (S Lock)**

SELECT \* FROM orders WITH (HOLDLOCK, ROWLOCK);

**Exclusive Lock (X Lock)**

UPDATE customers

SET balance = balance - 100

WHERE customer\_id = 1;

**Table Lock**

LOCK TABLE customers IN EXCLUSIVE MODE;

**Row Lock trong PostgreSQL**

SELECT \* FROM orders WHERE id = 1 FOR UPDATE;

**Deadlock Handling**

SET DEADLOCK\_PRIORITY HIGH;

**Khi nào cần sử dụng Lock?**

Khi có nhiều người dùng truy cập cùng một dữ liệu quan trọng (ngân hàng, thương mại điện tử).  
Khi cần đảm bảo tính nhất quán và tránh lỗi race condition.  
Khi xử lý giao dịch tài chính (banking transactions).  
 Khi thực hiện batch processing để cập nhật dữ liệu hàng loạt.