Các vấn đề xảy ra khi có truy xuất đồng thời lên cơ sở dữ liệu

* Mất dữ liệu cập nhật (Lost update)
  + Dữ liệu đọc không lặp lại (Non-repeatable read)
  + Bóng ma (Phantom read)
  + Đọc dữ liệu rác (Dirty read)

Có 3 kỹ thuật để điều khiển việc xử lý đồng thời: Locking, Timestamp, Validation

Chúng tôi chọn kỹ thuật locking để giải quyết các vấn đề này

Xử lý đồng thời trên bảng THANHVIEN

1. Lost update

Khi session1 thực hiện cập nhật bản ghi lên THANHVIEN có MATV = 'HVUIT001', session 2 thực hiện xóa THANHCVIEN HVUIT001 dẫn đến mất dữ liệu 🡺 Sử dụng khóa FOR UPDATE để khóa 2 session này lại, chỉ có session này thực hiện xong mới đến session kia

|  |  |  |
| --- | --- | --- |
| **T** | **Session 1** | **Session 2** |
|  | SET TRANSACTION ISOLATION LEVEL SERIALIZABLE; | SET TRANSACTION ISOLATION LEVEL SERIALIZABLE; |
|  | select \* from thanhvien FOR UPDATE; | select \* from thanhvien FOR UPDATE; |
|  |  | Delete \* from thanhvien  Where matv = 'HVUIT001'; |
|  | update thanhvien  set tentv = 'Nguyễn Thành Long'  where matv = 'HVUIT001'; |  |
|  |  | COMMIT; |
|  | COMMIT; |  |

1. Non – repeatable: session 1 thực hiện truy vấn đọc dữ liệu, nhưng dữ liệu này bị sửa đổi session 2, dẫn đến trả về giá trị khác so với lần trước đó 🡺 sử dụng khóa FOR UPDATE hoặc khóa FOR SHARE để đảm bảo tính toàn vẹn dữ liệu trong quá trình truy xuất.

|  |  |  |
| --- | --- | --- |
| **T** | **Session 1** | **Session 2** |
|  | SET TRANSACTION ISOLATION LEVEL SERIALIZABLE; | SET TRANSACTION ISOLATION LEVEL SERIALIZABLE; |
|  | select \* from thanhvien  where matv = 'HVUIT001' FOR UPDATE; |  |
|  |  | update thanhvien  set diachi = 'Ký Túc Xá Khu B, ĐHQG TPHCM'  where matv = 'HVUIT001'; |
|  | select \* from thanhvien  where matv = 'HVUIT001'; |  |
|  |  | COMMIT; |
|  | COMMIT; |  |

1. Phantom: session 1 thực hiện truy xuất dữ liệu, sau đó session 2 thực hiện xóa hoặc cập nhật lên dữ liệu đó, khi session 1 truy xuất lại thì thấy dữ liệu khác 🡺 sử dụng FOR SHARE

|  |  |  |
| --- | --- | --- |
| **T** | **Session 1** | **Session 2** |
|  | SET TRANSACTION ISOLATION LEVEL SERIALIZABLE; | SET TRANSACTION ISOLATION LEVEL SERIALIZABLE; |
|  | select \* from thanhvien  where matv like 'HVUIT%' FOR SHARE; |  |
|  |  | Insert into thanhvien(matv,tentv,madv) values ('HVUIT008', 'Trần Đăng Khoa', 'UIT'); |
|  | select \* from thanhvien  where matv like 'HVUIT%'; |  |
|  |  | COMMIT; |
|  | COMMIT; |  |

1. Dirty read: session 1 thực hienejd dọc dữ liệu từ 1 bản ghi, session 2 thực hiện cập nhật lên bản ghi đó, dẫn đến việc đọc dữ liệu đầu tiên của session1 không chính xác 🡺 Sử dụng transaction isolation level  để giứ cho dữ liệu luôn đọc đúng đến khi transaction kết thúc.

|  |  |  |
| --- | --- | --- |
| **T** | **Session 1** | **Session 2** |
|  | SET TRANSACTION ISOLATION LEVEL REPEATABLE READ; | BEGIN TRANSACTION; |
|  | select \* from thanhvien  where matv = 'HVUIT001'; |  |
|  |  | Update thanhvien  set tentv = 'Nguyễn Hoàng Long'  where matv = 'HVUIT001'; |
|  |  | COMMIT TRANSACTION: |
|  | COMMIT; |  |