**Họ và tên: Vũ Đức Huy**

**Lớp: DBI202 – AI1604.**

**Isolation Level**

**1) Định nghĩa**

**Isolation level** là một thuộc tính của transaction, qui định mức độ cô lập của dữ liệu mà transaction có thể truy nhập vào khi dữ liệu đó đang được cập nhật bởi một transaction khác. Khi một transaction cập nhật dữ liệu đang diễn ra, một phần dữ liệu sẽ bị thay đổi (ví dụ một số bản ghi của bảng được sửa đổi hoặc bị xóa bỏ, một số được thêm mới), vậy các transaction hoặc truy vấn khác xảy ra đồng thời và cùng tác động vào các bản ghi đó sẽ diễn ra thế nào? Chúng sẽ phải đợi đến khi transaction đầu hoàn thành hay có thể thực hiện song song, kết quả dữ liệu nhận được là trong khi hay sau khi cập nhật? Bạn có thể điều khiển những hành vi này thông qua việc đặt isolation level của từng transaction. SQL Server cung cấp các mức isolation level sau xếp theo thứ tự tăng dần của mức độ cô lập của dữ liệu: Read Uncommitted, Read Commited, Repeatable Read, và Serializable. Từ bản 2005 bắt đầu bổ sung thêm một loại mới là Snapshot. Trong Sql mặc định là Read commited nếu không quan tâm đến transaction.

Tuỳ các cấp độ thì sẽ có các rủi ro nhất định.Sẽ có **3** rủi ro chính đó là:

・ **Uncommitted dependency (dirty read):** Tác động này xảy ra khi 1 transaction A đọc 1 row khi nó đang được update bởi 1 transaction B khác và chưa được commit. Transaction A sẽ đọc dữ liệu vẫn chưa được commit. Ví dụ 1 người A trong tài khoản có 3 triệu và đang thực hiện 1 giao dịch để nạp vào 2 triệu cho tài khoản. Transaction đã chạy update xong tài khoản lên 5 triệu nhưng vẫn còn 1 số thao tác chưa chạy tới và dữ liệu này vẫn chưa được commit. 1 người B khác dùng chung tài khoản thực hiện kiểm tra số dư, và kết quả trả về là 5 triệu. Giao dịch của người A xảy ra sự cố và báo lỗi, dữ liệu được roll back về 3 triệu. Như vậy người B đang nhận được dữ liệu sai.

・**Inconsistent analysis (nonrepeatable read):** Trường hợp này xảy ra khi 1 transaction A đọc 1 đơn vị dữ liệu nhiều lần và kết quả khác nhau giữa các lần do giữa thời gian đọc của các lân đó, dữ liệu bị 1 transaction khác commit thay đổi.

・**Phantom reads**: xảy ra khi 2 queries giống hệt nhau được thực hiện nhưng list rows kết quả trả về lại khác nhau. Ví dụ, khi mua vé xem phim. 2 câu lệnh mua vé . Câu lệnh 1 mua vé 1. Câu lệnh 2 mua vé 2 thì 2 dữ liệu được xuất ra sẽ khác nhau.

**2)Các mức độ của Isolation**

**a. Read Uncommitted**

**Đặc điểm:**

– Không thiết lập Shared Lock trên những đơn vị dữ liệu cần đọc. Do đó không phải chờ khi đọc dữ liệu (kể cả khi dữ liệu đang bị lock bởi giao tác khác)

– Vẫn tạo Exclusive Lock trên đơn vị dữ liệu được ghi, Exclusive Lock được giữ cho đến hết giao tác

**Ưu điểm:**

– Tốc độ xử lý rất nhanh

– Không cản trở những giao tác khác thực hiện việc cập nhật dữ liệu

**Nhược điểm:**

– Có khả năng xảy ra mọi vấn đề khi xử lý đồng thời :

+ Dirty Reads

+ Unrepeatable Reads

+ Phantoms

+ Lost Updates

**b. Read Committed**

**Đặc điểm:**

– Đây là mức độ cô lập mặc định của SQL Server

– Tạo Shared Lock trên đơn vị dữ liệu được đọc, Shared Lock được giải phóng ngay sau khi đọc xong dữ liệu

– Tạo Exclusive Lock trên đơn vị dữ liệu được ghi, Exclusive Lock được giữ cho đến hết giao tác

**Ưu điểm:**

– Giải quyết vấn đề Dirty Reads

– Shared Lock được giải phóng ngay, không cần phải giữ cho đến hết giao tác nên không cản trở nhiều đến thao tác cập nhật của các giao tác khác.

**Nhược điểm:**

– Chưa giải quyết được vấn đề Unrepeatable Reads, Phantoms,Lost Updates

– Phải chờ nếu đơn vị dữ liệu cần đọc đang được giữ khoá ghi (xlock)

**c. Repeatable Read**

**Đặc điểm:**

– Tạo Shared Lock trên đơn vị dữ liệu được đọc và giữ shared lock này đến hết giao tác => Các giao tác khác phải chờ đến khi giao tác này kết thúc nếu muốn cập nhật, thay đổi giá trị trên đơn vị dữ liệu này .

– (Repeatable Read = Read Committed + Giải quyết Unrepeatable Reads)

– Tạo Exclusive Lock trên đơn vị dữ liệu được ghi, Exclusive Lock được giữ cho đến hết giao tác.

**Ưu điểm:**

– Giải quyết vấn đề Dirty Reads và Unrepeatable Reads

**Nhược điểm:**

– Chưa giải quyết được vấn đề Phantoms, do vẫn cho phép insert những dòng dữ liệu thỏa điều kiện thiết lập shared lock

– Phải chờ nếu đơn vị dữ liệu cần đọc đang được giữ khoá ghi (xlock)

– Shared lock được giữ đến hết giao tác ==> cản trở việc cập nhật dữ liệu của các giao tác khác

**d. Serializable**

**Đặc điểm:**

– Tạo Shared Lock trên đơn vị dữ liệu được đọc và giữ shared lock này đến hết giao tác => Các giao tác khác phải chờ đến khi giao tác này kết thúc nếu muốn cập nhật, thay đổi giá trị trên đơn vị dữ liệu này .

– Không cho phép Insert những dòng dữ liệu thỏa mãn điều kiện

thiết lập Shared Lock (sử dụng Key Range Lock) ==> Serializable = Repeatable Read + Giải quyết Phantoms

– Tạo Exclusive Lock trên đơn vị dữ liệu được ghi, Exclusive Lock

được giữ cho đến hết giao tác.

**Ưu điểm:**

– Giải quyết thêm được vấn đề Phantoms

**Nhược điểm:**

– Phải chờ nếu đơn vị dữ liệu cần đọc đang được giữ khoá ghi (xlock)

– Cản trở nhiều đến việc cập nhật dữ liệu của các giao tác khác.

**Bảng rủi ro của các level**

|  |  |  |  |
| --- | --- | --- | --- |
| **Isolation Level** | **Dirty Read** | **Non Repeatable** | **Phantom** |
| Read Uncommitted | Yes | Yes | Yes |
| Read Commited | No | Yes | Yes |
| Repeatable Read | No | No | Yes |
| Serializable | No | No | No |