

## Phần III

# Quản lý giao dịch Oracle

Quản lý giao dịch là việc sử dụng các giao dịch để đảm bảo tính đồng thời và nhất quán của dữ liệu.

Phần này gồm các chương sau:

- [Tính đồng thời và nhất quán của dữ liệu](#)
- [Giao dịch](#)

## 9

# Tính đồng thời và nhất quán của dữ liệu

Chương này giải thích cách Cơ sở dữ liệu Oracle duy trì dữ liệu nhất quán trong môi trường cơ sở dữ liệu nhiều người dùng.

Chương này có các phần sau:

- Giới thiệu về tính đồng thời và nhất quán của dữ liệu
- Tổng quan về các mức cô lập giao dịch cơ sở dữ liệu Oracle
- Tổng quan về Cơ chế khóa cơ sở dữ liệu Oracle
- Tổng quan về Khóa tự động
- Tổng quan về khóa dữ liệu thủ công
- Tổng quan về Khóa do người dùng xác định

## Giới thiệu về tính đồng thời và nhất quán của dữ liệu

Trong cơ sở dữ liệu một người dùng, người dùng có thể sửa đổi dữ liệu mà không cần quan tâm đến việc những người dùng khác sửa đổi cùng một dữ liệu cùng lúc. Tuy nhiên, trong cơ sở dữ liệu nhiều người dùng, các câu lệnh trong nhiều giao dịch đồng thời có thể cập nhật cùng một dữ liệu. Các giao dịch được thực hiện đồng thời phải tạo ra kết quả có ý nghĩa và nhất quán.

Cơ sở dữ liệu nhiều người dùng phải cung cấp những thông tin sau:

- Sự đảm bảo rằng người dùng có thể truy cập dữ liệu cùng một lúc ([dữ liệu đồng thời](#))
- Sự đảm bảo rằng mỗi người dùng đều nhìn thấy một cái nhìn nhất quán về dữ liệu ([dữ liệu tính nhất quán](#)), bao gồm những thay đổi có thể nhìn thấy được thực hiện bởi các giao dịch của chính người dùng và các giao dịch đã cam kết của những người dùng khác

Để mô tả hành vi giao dịch nhất quán khi các giao dịch chạy đồng thời, các nhà nghiên cứu cơ sở dữ liệu đã xác định một mô hình cách ly giao dịch được gọi là [khả năng tuần tự hóa](#).

Một giao dịch có thể tuần tự hóa hoạt động trong một môi trường khiến nó có vẻ như không có người dùng nào khác đang sửa đổi dữ liệu trong cơ sở dữ liệu.

Mặc dù mức độ cách ly giữa các giao dịch này thường được mong muốn nhưng việc chạy nhiều ứng dụng ở chế độ tuần tự hóa có thể ảnh hưởng nghiêm trọng đến thông lượng ứng dụng. Việc cách ly hoàn toàn các giao dịch đang chạy đồng thời có thể có nghĩa là một giao dịch không thể thực hiện việc chen vào bảng đang được giao dịch khác truy vấn. Nói tóm lại, những cân nhắc trong thế giới thực thường đòi hỏi sự thỏa hiệp giữa sự cô lập và hiệu suất giao dịch hoàn hảo.

Cơ sở dữ liệu Oracle duy trì tính nhất quán của dữ liệu bằng cách sử dụng [mô hình nhất quán đa phiên bản](#) và nhiều loại khóa và giao dịch khác nhau. Bằng cách này, cơ sở dữ liệu có thể hiển thị chế độ xem dữ liệu cho nhiều người dùng đồng thời, với mỗi chế độ xem nhất quán tại một thời điểm. Bởi vì các phiên bản khác nhau của khối dữ liệu có thể tồn tại đồng thời nên các giao dịch có thể đọc phiên bản dữ liệu được cam kết tại thời điểm mà truy [vấn yêu cầu](#) và trả về kết quả nhất quán tại một thời điểm.



Xem thêm:

"Tính toàn vẹn dữ liệu" và "Giao dịch "

## Tính nhất quán đọc đa phiên bản

Trong Cơ sở dữ liệu Oracle, đa phiên bản là khả năng hiện thực hóa đồng thời nhiều phiên bản dữ liệu. Cơ sở dữ liệu Oracle duy trì tính nhất quán khi đọc đa phiên bản.

Các truy vấn của cơ sở dữ liệu Oracle có các đặc điểm sau:

- Truy vấn nhất quán đọc

Dữ liệu được truy vấn trả về được cam kết và nhất quán trong một thời điểm.



Ghi chú:

Cơ sở dữ liệu Oracle không bao giờ cho phép **đọc sai**, xảy ra khi một giao dịch đọc dữ liệu không được cam kết trong một giao dịch khác.

Để minh họa vấn đề với việc đọc sai, giả sử một giao dịch cập nhật giá trị cột mà không cam kết. Giao dịch thứ hai đọc giá trị cập nhật và giá trị bản (không được cam kết). Phiên đầu tiên khôi phục giao dịch để cột có giá trị cũ, nhưng giao dịch thứ hai tiếp tục sử dụng giá trị được cập nhật, làm hỏng cơ sở dữ liệu. Đọc bản làm tổn hại đến **tính toàn vẹn của dữ liệu**, vi phạm khóa ngoại và bỏ qua các ràng buộc duy nhất.

- Truy vấn không chặn

Người đọc và người ghi dữ liệu không chặn nhau.



Xem thêm:

"Tóm tắt về hành vi khóa"

## Tính nhất quán khi đọc ở cấp độ câu lệnh

Cơ sở dữ liệu Oracle luôn thực thi tính nhất quán khi đọc ở cấp độ câu lệnh, điều này đảm bảo rằng dữ liệu được trả về bởi một truy vấn duy nhất được cam kết và nhất quán tại một thời điểm.

Thời điểm mà một câu lệnh SQL duy nhất nhất quán phụ thuộc vào mức độ cô lập giao dịch và bản chất của truy vấn:

- Ở mức độ cô lập đã cam kết đã đọc, thời điểm này là thời điểm câu lệnh được mở. Ví dụ: nếu câu lệnh SELECT mở ở SCN 1000 thì câu lệnh này nhất quán với SCN 1000.

- Trong giao dịch có thể tuần tự hóa hoặc chỉ đọc, thời điểm này là thời điểm giao dịch bắt đầu. Ví dụ: nếu một giao dịch bắt đầu ở SCN 1000 và nếu nhiều câu lệnh CHỌN xảy ra trong giao dịch này thì mỗi câu lệnh sẽ nhất quán với SCN 1000.
- Trong thao tác Truy vấn Flashback (CHỌN ... NHƯ VẬY), câu lệnh CHỌN chỉ định rõ ràng thời điểm. Ví dụ: bạn có thể truy vấn một bảng như nó xuất hiện lúc 2 giờ chiều Thứ Năm tuần trước



Xem thêm:

Hướng dẫn phát triển cơ sở dữ liệu Oracle để tìm hiểu về Flashback Query

## Tính nhất quán khi đọc ở cấp độ giao dịch

Cơ sở dữ liệu Oracle cũng có thể cung cấp tính nhất quán khi đọc cho tất cả các truy vấn trong một giao dịch, được gọi là tính nhất quán khi đọc ở cấp độ giao dịch.

Trong trường hợp này, mỗi câu lệnh trong giao dịch sẽ xem dữ liệu từ cùng một thời điểm. Đây là thời điểm giao dịch bắt đầu.

Các truy vấn được thực hiện bởi một giao dịch có thể tuần tự hóa sẽ thấy những thay đổi do chính giao dịch đó thực hiện. Ví dụ: một giao dịch cập nhật nhân viên và sau đó truy vấn nhân viên sẽ thấy các cập nhật. Tính nhất quán khi đọc ở cấp độ giao dịch tạo ra các lần đọc lặp lại và không hiển thị truy vấn cho các lần đọc ảo.

## Đọc tính nhất quán và hoàn tác các phân đoạn

Để quản lý mô hình nhất quán đọc đa phiên bản, cơ sở dữ liệu phải tạo một tập dữ liệu nhất quán đọc khi một bảng được truy vấn và cập nhật đồng thời.

Cơ sở dữ liệu Oracle đạt được tính nhất quán khi đọc thông qua [việc hoàn tác dữ liệu](#).

Bất cứ khi nào người dùng sửa đổi dữ liệu, Cơ sở dữ liệu Oracle sẽ tạo các mục hoàn tác, nó ghi vào các phân đoạn hoàn tác. Các phân đoạn hoàn tác chứa các giá trị cũ của dữ liệu đã bị thay đổi bởi các giao dịch chưa được cam kết hoặc đã được cam kết gần đây. Do đó, nhiều phiên bản của cùng một dữ liệu, tại các thời điểm khác nhau, có thể tồn tại trong cơ sở dữ liệu. Cơ sở dữ liệu có thể sử dụng ảnh chụp nhanh dữ liệu tại các thời điểm khác nhau để cung cấp chế độ xem dữ liệu nhất quán khi đọc và cho phép truy vấn không chặn.

Tính nhất quán khi đọc được đảm bảo trong môi trường phiên bản đơn và môi trường Cụm ứng dụng thực của Oracle (Oracle RAC). Oracle RAC sử dụng cơ chế chuyển khối đệm từ bộ đệm này sang bộ đệm khác được gọi là kết hợp bộ đệm để truyền các hình ảnh nhất quán về khả năng đọc của các khối dữ liệu từ phiên bản cơ sở dữ liệu này sang phiên bản cơ sở dữ liệu khác.



Xem thêm:

- "Hoàn tác phân đoạn"

để tìm hiểu về cách hoàn tác bộ nhớ

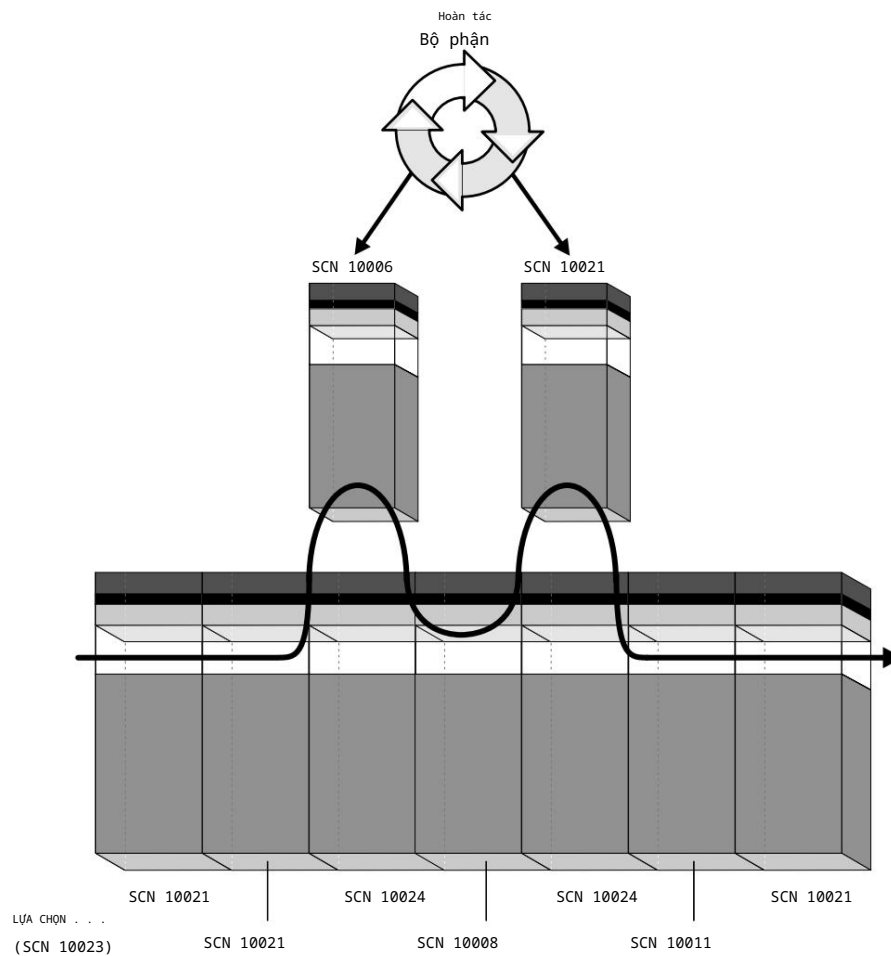
- "LOB nội bộ" để tìm hiểu về cơ chế nhất quán đọc cho LOB

- Cơ sở dữ liệu Oracle 2 ngày + Hướng dẫn về cụm ứng dụng thực để tìm hiểu về hợp nhất bộ đệm

## Đọc tính nhất quán: Ví dụ

Ví dụ này hiển thị một truy vấn sử dụng dữ liệu hoàn tác để cung cấp tính nhất quán khi đọc ở cấp câu lệnh ở cấp độ cách ly đã cam kết đọc.

Hình 9-1 Đọc nhất quán ở mức cô lập đã cam kết đọc



Khi cơ sở dữ liệu truy xuất các khối dữ liệu thay mặt cho một truy vấn, cơ sở dữ liệu sẽ đảm bảo rằng dữ liệu trong mỗi khối phản ánh nội dung của khối khi truy vấn bắt đầu. Các

cơ sở dữ liệu khôi phục các thay đổi đối với khối khi cần thiết để xây dựng lại khối tại thời điểm truy vấn bắt đầu xử lý.

Cơ sở dữ liệu sử dụng cơ chế đặt hàng nội bộ được gọi là **SCN** để đảm bảo thứ tự giao dịch. Khi câu lệnh **SELECT** bước vào giai đoạn thực thi, cơ sở dữ liệu sẽ xác định SCN được ghi tại thời điểm truy vấn bắt đầu thực thi. Trong **Hình 9-1**, SCN này là 10023. Truy vấn chỉ nhìn thấy dữ liệu đã cam kết liên quan đến SCN 10023.

Trong **Hình 9-1**, các khối có SCN sau 10023 biểu thị dữ liệu đã thay đổi, như được hiển thị bởi hai khối có SCN 10024. Câu lệnh **SELECT** yêu cầu một phiên bản của khối nhất quán với các thay đổi đã cam kết. Cơ sở dữ liệu sao chép các khối dữ liệu hiện tại vào bộ đệm mới và áp dụng dữ liệu hoàn tác để xây dựng lại các phiên bản trước của khối.

Các khối dữ liệu được xây dựng lại này được gọi là bản sao đọc nhất quán (CR).

Trong **Hình 9-1**, cơ sở dữ liệu tạo hai bản sao CR: một khối phù hợp với SCN 10006 và khối còn lại phù hợp với SCN 10021. Cơ sở dữ liệu trả về dữ liệu được xây dựng lại cho truy vấn. Bằng cách này, Cơ sở dữ liệu Oracle ngăn chặn việc đọc sai.



Xem thêm:

"Bộ đệm cơ sở dữ liệu" và "Số thay đổi hệ thống (SCN)"

## Độc danh sách giao dịch nhất quán và quan tâm

Tiêu đề khối của mỗi khối phân đoạn chứa danh sách giao dịch quan tâm (ITL).

Cơ sở dữ liệu sử dụng ITL để xác định xem giao dịch có được thực hiện hay không khi cơ sở dữ liệu bắt đầu sửa đổi khối.

Các mục trong ITL mô tả giao dịch nào có hàng bị khóa và hàng nào trong khối chứa các thay đổi đã cam kết và không được cam kết. ITL trở tới bảng giao dịch trong phân đoạn hoàn tác, cung cấp thông tin về thời gian thực hiện các thay đổi đối với cơ sở dữ liệu.

Theo một nghĩa nào đó, tiêu đề khối chứa lịch sử giao dịch gần đây ảnh hưởng đến từng hàng trong khối. Tham số **INITRANS** của câu lệnh **CREATE TABLE** và **ALTER TABLE** kiểm soát lượng lịch sử giao dịch được lưu giữ.



Xem thêm:

Tham khảo ngôn ngữ SQL cơ sở dữ liệu Oracle để tìm hiểu về tham số **INITRANS**

## Cơ chế khóa

Nói chung, cơ sở dữ liệu nhiều người dùng sử dụng một số dạng khóa dữ liệu để giải quyết các vấn đề liên quan đến tính đồng thời, tính nhất quán và tính toàn vẹn của dữ liệu.

Khóa là một cơ chế ngăn chặn sự tương tác phá hoại giữa các giao dịch truy cập vào cùng một tài nguyên.



Xem thêm:

"Tổng quan về Cơ chế khóa cơ sở dữ liệu Oracle"

## Mức cô lập giao dịch ANSI/ISO

Tiêu chuẩn SQL đã được cả ANSI và ISO/IEC áp dụng, xác định bốn mức độ cô lập giao dịch. Các mức độ này có mức độ tác động khác nhau thông lượng xử lý giao dịch.

Các mức độ cô lập này được xác định theo các hiện tượng phải được ngăn chặn giữa các giao dịch thực hiện đồng thời. Các hiện tượng có thể phòng ngừa được là:

- Đọc bản

Một giao dịch đọc dữ liệu được ghi bởi một giao dịch khác chưa được ghi đã cam kết chưa.

- Số lần đọc không lặp lại (mờ)

Một giao dịch đọc lại dữ liệu mà nó đã đọc trước đó và thấy rằng một giao dịch khác đã cam kết giao dịch đã sửa đổi hoặc xóa dữ liệu. Ví dụ: người dùng truy vấn một hàng và sau đó truy vấn cùng một hàng, chỉ để phát hiện ra rằng dữ liệu đã thay đổi.

- Đọc ma

Một giao dịch chạy lại một truy vấn trả về một tập hợp các hàng thỏa mãn tìm kiếm điều kiện và thấy rằng một giao dịch đã cam kết khác đã chèn thêm các hàng thỏa mãn điều kiện.

Ví dụ: một giao dịch truy vấn số lượng nhân viên. Năm phút sau nó thực hiện cùng một truy vấn, nhưng bây giờ số lượng đã tăng thêm một vì một người dùng khác đã chèn một bản ghi cho nhân viên mới được thuê. Nhiều dữ liệu hơn thỏa mãn tiêu chí truy vấn hơn trước, nhưng không giống như đọc mờ, dữ liệu đã đọc trước đó không thay đổi.

Tiêu chuẩn SQL xác định bốn mức độ cô lập theo các hiện tượng mà một giao dịch chạy ở một mức độ cô lập cụ thể được phép trải nghiệm. [Bảng 9-1](#) cho thấy các cấp độ.

Bảng 9-1 Hiện tượng đọc có thể phòng ngừa được theo mức cô lập

Mức độ cách ly	Đọc bản	Không thể lặp lại Đọc	Đọc ảo
Đọc không cam kết	Khả thi	Khả thi	Khả thi
Đọc cam kết	Không thể	Khả thi	Khả thi
Đọc lặp lại	Không thể	Không thể	Khả thi
Có thể tuần tự hóa	Không thể	Không thể	Không thể

Cơ sở dữ liệu Oracle cung cấp các mức cách ly được cam kết đọc (mặc định) và có thể tuần tự hóa. Ngoài ra, cơ sở dữ liệu còn cung cấp chế độ chỉ đọc.

**Xem thêm:**

- "[Tổng quan về các mức cô lập giao dịch cơ sở dữ liệu Oracle](#)" để tìm hiểu về các mức cô lập đã cam kết đọc, có thể tuần tự hóa và chỉ đọc
- Tài liệu tham khảo ngôn ngữ SQL của cơ sở dữ liệu Oracle để thảo luận về Oracle Sự phù hợp của cơ sở dữ liệu với các tiêu chuẩn SQL

## Tổng quan về các mức cô lập giao dịch cơ sở dữ liệu Oracle

Tiêu chuẩn ANSI cho các mức cách ly giao dịch được xác định theo các hiện tượng được phép hoặc ngăn chặn đối với từng mức cách ly.

Cơ sở dữ liệu Oracle cung cấp các mức cách ly giao dịch:

- [Đọc mức cô lập đã cam kết](#)
- [Mức cô lập có thể tuần tự hóa](#)
- [Mức cô lập chỉ đọc](#)

**Xem thêm:**

- [Hướng dẫn phát triển cơ sở dữ liệu Oracle](#) để tìm hiểu thêm về mức độ cô lập giao dịch
- Cơ sở dữ liệu Oracle Tham chiếu ngôn ngữ SQL và Cơ sở dữ liệu Oracle Tham khảo ngôn ngữ PL/SQL để tìm hiểu về SET TRANSACTION ISOLATION MỨC ĐỘ

### Đọc mức cô lập đã cam kết

Ở cấp độ cách ly đã cam kết đọc, mọi truy vấn được thực hiện bởi một giao dịch chỉ thấy dữ liệu được cam kết trước khi truy vấn—không phải giao dịch—bắt đầu.

Mức cô lập này là mặc định. Nó phù hợp với môi trường cơ sở dữ liệu trong đó có ít giao dịch có khả năng xung đột.

Truy vấn trong giao dịch đã cam kết đọc sẽ tránh đọc dữ liệu đã cam kết trong khi truy vấn đang được tiến hành. Ví dụ: nếu một truy vấn đang quét một nửa bảng hàng triệu và nếu một giao dịch khác thực hiện cập nhật cho hàng 950.000 thì truy vấn sẽ không thấy thay đổi này khi đọc hàng 950.000. Tuy nhiên, do cơ sở dữ liệu không ngăn các giao dịch khác sửa đổi dữ liệu được truy vấn đọc nên các giao dịch khác có thể thay đổi dữ liệu giữa các lần thực hiện truy vấn. Do đó, một giao dịch chạy cùng một truy vấn hai lần có thể gặp phải tình trạng đọc mờ và ảo.

### Đọc nhất quán ở cấp độ cô lập đã cam kết đọc

Cơ sở dữ liệu cung cấp một tập kết quả nhất quán cho mọi truy vấn, đảm bảo tính nhất quán của dữ liệu mà người dùng không cần thực hiện hành động nào.



Một **truy vấn ngầm**, chẳng hạn như truy vấn được ngụ ý bởi mệnh đề WHERE trong câu lệnh UPDATE, là đảm bảo một tập hợp các kết quả nhất quán. Tuy nhiên, mỗi câu lệnh trong một truy vấn ngầm không nhìn thấy những thay đổi được thực hiện bởi chính câu lệnh DML, nhưng nhìn thấy dữ liệu như nó tồn tại trước khi những thay đổi được thực hiện.

Nếu danh sách SELECT chứa hàm PL/SQL thì cơ sở dữ liệu sẽ áp dụng mức câu lệnh đọc tính nhất quán ở cấp câu lệnh để chạy SQL trong mã hàm PL/SQL, chứ không phải ở cấp độ SQL gốc. Ví dụ: một hàm có thể truy cập vào một bảng dữ liệu của nó được thay đổi và cam kết bởi người dùng khác. Đối với mỗi lần thực hiện SELECT trong hàm này, một ảnh chụp nhanh mới nhất quán về đọc sẽ được thiết lập.



Xem thêm:

"Truy vấn phụ"

## Các bản ghi xung đột trong các giao dịch đã cam kết đã đọc

Trong một giao dịch đã cam kết đọc, một giao dịch ghi xung đột xảy ra khi giao dịch cố gắng thay đổi một hàng được cập nhật bởi một giao dịch đồng thời không được cam kết.

Giao dịch ngăn chặn việc sửa đổi hàng đôi khi được gọi là giao dịch chặn. giao dịch. Giao dịch đã cam kết đã đọc chờ giao dịch chặn kết thúc và nhà khóa hàng của nó.

Các tùy chọn như sau:

- Nếu giao dịch chặn quay trở lại thì giao dịch đang chờ sẽ tiếp tục thay đổi hàng đã khóa trước đó như thể giao dịch kia chưa từng tồn tại.
- Nếu giao dịch chặn hoàn tất và giải phóng khóa của nó thì thời gian chờ đợi giao dịch tiến hành cập nhật theo dự định của nó đối với hàng mới được thay đổi.

Bảng sau đây cho thấy giao dịch 1, có thể được tuần tự hóa hoặc được đọc như thế nào đã cam kết, tương tác với giao dịch đã cam kết đã đọc 2. Nó hiển thị một tình huống cổ điển được gọi là **bản cập nhật bị mất**. Bản cập nhật được thực hiện bởi giao dịch 1 thậm chí không có trong bảng mặc dù giao dịch 1 đã cam kết nó. Việc đưa ra một chiến lược để xử lý các bản cập nhật bị mất là một phần quan trọng của việc phát triển ứng dụng.

Bảng 9-2 Xung đột ghi và cập nhật bị mất trong giao dịch READ CAM KẾT

Phiên 1	Buổi 2	Giải trình						
<pre>SQL&gt; CHON họ, lương TỪ nhân viên Ở ĐẦU họ_TRÊN ('Banda', 'Greene', 'Gợi ý');</pre> <table><tr><th>HỌ</th><th>LƯƠNG</th></tr><tr><td>Banda</td><td>6200</td></tr><tr><td>Greene</td><td>9500</td></tr></table>	HỌ	LƯƠNG	Banda	6200	Greene	9500	Không có hành động.	Phần 1 truy vấn mức lương của Banda, Greene và Hintz. KHÔNG nhân viên tên Hintz được tìm thấy.
HỌ	LƯƠNG							
Banda	6200							
Greene	9500							

Bảng 9-2 (Tiếp theo) Xung đột ghi và cập nhật bị mất trong giao dịch READ CAM KẾT

Phiên 1	Buổi 2	Giải trình
SQL> CẬP NHẬT nhân viên Mức lương SET = 7000 WHERE Last_name = 'Banda';	Không có hành động.	Phiên 1 bắt đầu giao dịch bằng cập nhật mức lương Banda. Các mức cô lập mặc định cho giao dịch 1 được ĐỌC CAM KẾT.
Không có hành động.	SQL> THIẾT LẬP GIAO DỊCH MỨC ĐỘ GIẢI QUYẾT ĐỌC CAM KẾT;	Phiên 2 bắt đầu giao dịch 2 và đặt mức cách ly rõ ràng là ĐỌC CAM KẾT.
Không có hành động.	SQL> CHON họ, lương TỪ nhân viên Ở ĐẦU họ_TRÊN ('Banda', 'Greene', 'Gợi ý');  HỌ                    LƯƠNG ----- Banda                6200 Greene               9500	Giao dịch 2 truy vấn mức lương cho Banda, Greene và Hintz. Cơ sở dữ liệu Oracle sử dụng chức năng đọc tính nhất quán để hiển thị mức lương cho Banda trước khi không cam kết cập nhật được thực hiện bởi giao dịch 1.
Không có hành động.	SQL> CẬP NHẬT nhân viên Mức lương BỘ = 9900 Ở ĐẦU Last_name='Greene';	Giao dịch 2 cập nhật lương cho Greene thành công vì giao dịch 1 chỉ bị khóa Hàng Banda.
SQL> XÁC NHẬN VÀO nhân viên (employee_id, họ_tên, email, ngày thuê, job_id)  GIÁ TRỊ (210, 'Hintz', 'JHINTZ', SYSDATE, 'SH_CLERK');	Không có hành động.	Giao dịch 1 chèn một hàng cho nhân viên Hintz, nhưng không làm.
Không có hành động.	SQL> CHON họ, lương TỪ nhân viên Ở ĐẦU họ_TRÊN ('Banda', 'Greene', 'Gợi ý');  HỌ                    LƯƠNG ----- Banda                6200 Greene               9900	Giao dịch 2 truy vấn mức lương cho nhân viên Banda, Greene, và Hintz.  Giao dịch 2 thấy bản cập nhật của chính nó tới mức lương của Greene. Giao dịch 2 không thấy cập nhật không cam kết về mức lương cho Banda hoặc phần chèn cho Hintz được thực hiện bởi giao dịch 1.
Không có hành động.	SQL> CẬP NHẬT nhân viên SET Lương = 6300 WHERE họ_ = 'Banda';  -- lời nhắc không trả về	Giao dịch 2 cố gắng cập nhật hàng dành cho Banda, đó là hiện đang bị khóa bởi giao dịch 1, tạo ra một văn bản xung đột. Giao dịch 2 đợi cho đến khi giao dịch 1 kết thúc.
SQL> CAM KẾT;	Không có hành động.	Giao dịch 1 cam kết công việc của nó, kết thúc giao dịch.

Phiên 1	Buổi 2	Giải trình								
Không có hành động.	<p>Đã cập nhật 1 hàng.</p> <p>SQL&gt;</p>	<p>Khóa trên hàng Banda bây giờ là được phát hành, vì vậy giao dịch 2 tiến hành cập nhật nó lên lương cho Banda.</p>								
Không có hành động.	<p>SQL&gt; CHON họ, lương TỪ nhân viên Ở ĐẦU họ_TRÊN ( 'Banda', 'Greene', 'Gợi ý');</p> <table> <tr> <td>HỌ</td> <td>LƯƠNG</td> </tr> <tr> <td>Banda</td> <td>6300</td> </tr> <tr> <td>Greene</td> <td>9900</td> </tr> <tr> <td>gợi ý</td> <td></td> </tr> </table>	HỌ	LƯƠNG	Banda	6300	Greene	9900	gợi ý		<p>Giao dịch 2 truy vấn mức lương cho nhân viên Banda, Greene, và Hintz. Phần chèn Hintz được cam kết bởi giao dịch 1 bây giờ là hiển thị cho giao dịch 2.</p> <p>Giao dịch 2 thấy bản cập nhật của chính nó đến mức lương của Banda.</p>
HỌ	LƯƠNG									
Banda	6300									
Greene	9900									
gợi ý										
Không có hành động.	LÀM;	<p>Giao dịch 2 cam kết công việc của nó, kết thúc giao dịch.</p>								
<p>SQL&gt; CHON họ, lương TỪ nhân viên Ở ĐẦU họ_TRÊN ( 'Banda', 'Greene', 'Gợi ý');</p> <table> <tr> <td>HỌ</td> <td>LƯƠNG</td> </tr> <tr> <td>Banda</td> <td>6300</td> </tr> <tr> <td>Greene</td> <td>9900</td> </tr> <tr> <td>gợi ý</td> <td></td> </tr> </table>	HỌ	LƯƠNG	Banda	6300	Greene	9900	gợi ý		<p>Không có hành động.</p>	<p>Phiên 1 truy vấn các hàng cho Banda, Greene và Hintz. Các lương của Banda là 6300, tức là cập nhật được thực hiện bởi giao dịch 2. Cập nhật lương của Banda lên 7000 được thực hiện bởi giao dịch 1 là nay đã "mất".</p>
HỌ	LƯƠNG									
Banda	6300									
Greene	9900									
gợi ý										

Xem thêm:

- "Sử dụng Khóa" để tìm hiểu về các bản cập nhật bị mất
- "Khóa hàng (TX)" để tìm hiểu thời điểm và lý do cơ sở dữ liệu nhận được khóa hàng

Ở mức cô lập có thể tuân tự hóa, một giao dịch chỉ nhìn thấy những thay đổi được thực hiện tại thời gian giao dịch—không phải truy vấn—bắt đầu và những thay đổi được thực hiện bởi giao dịch chính nó.

Một giao dịch có thể tuân tự hóa hoạt động trong một môi trường khiến nó có vẻ như không có những người dùng khác đang sửa đổi dữ liệu trong cơ sở dữ liệu. Cách ly tuân tự hóa phù hợp cho môi trường:

- Với cơ sở dữ liệu lớn và các giao dịch ngắn chỉ cập nhật một vài hàng

- Trường hợp có khả năng hai giao dịch đồng thời sửa đổi cùng một hàng tương đối thấp
- Trường hợp các giao dịch tương đối dài chủ yếu chỉ được đọc

Trong sự cô lập có thể tuần tự hóa, tính nhất quán khi đọc thường đạt được ở cấp độ câu lệnh mở rộng cho toàn bộ giao dịch. Bất kỳ hàng nào được giao dịch đọc đều được đảm bảo là tương tự khi đọc lại. Mọi truy vấn đều được đảm bảo trả về kết quả tương tự trong suốt thời gian của giao dịch, do đó những thay đổi được thực hiện bởi các giao dịch khác sẽ không được hiển thị đối với truy vấn bất kể nó đã chạy được bao lâu. Các giao dịch tuần tự hóa không trải nghiệm đọc không rõ ràng, đọc mờ hoặc đọc ảo.

Cơ sở dữ liệu Oracle cho phép một giao dịch có thể tuần tự hóa chỉ sửa đổi một hàng nếu thay đổi thành hàng được thực hiện bởi các giao dịch khác đã được cam kết khi có thể tuần tự hóa giao dịch bắt đầu. Cơ sở dữ liệu tạo ra lỗi khi một giao dịch có thể tuần tự hóa cố gắng cập nhật hoặc xóa dữ liệu được thay đổi bởi một giao dịch khác được thực hiện sau giao dịch tuần tự hóa đã bắt đầu:

ORA-08177: Không thể tuần tự hóa quyền truy cập cho giao dịch này

Khi một giao dịch có thể tuần tự hóa không thành công với lỗi ORA-08177, ứng dụng có thể mất một số hành động, trong đó có những hành động sau:

- Cam kết công việc thực hiện đến thời điểm đó
- Thực thi các câu lệnh bổ sung (nhưng khác), có thể sau khi quay trở lại điểm lưu trữ được thiết lập trước đó trong giao dịch
- Khôi phục toàn bộ giao dịch

Bảng sau đây cho thấy một giao dịch có thể tuần tự hóa tương tác với các giao dịch khác như thế nào giao dịch. Nếu giao dịch có thể tuần tự hóa không cố gắng thay đổi một hàng được cam kết bởi một giao dịch khác sau khi giao dịch có thể tuần tự hóa bắt đầu, sau đó là một truy cập được tuần tự hóa vấn đề được tránh.

Bảng 9-3 Giao dịch tuần tự hóa

Phiên 1	Buổi 2	Giải trình
<div>SQL&gt; CHON họ, lương</div> <div>Từ nhân viên ở ĐẦU</div> <div>họ_TRÊN ('Banda',</div> <div>'Greene', 'Hintz');</div> <div>HỌ</div> <div>LƯƠNG</div> <div>Banda 6200</div> <div>Greene 9500</div>	Không có hành động.	Phần 1 truy vấn mức lương của Banda, Greene và Hintz. KHÔNG nhân viên tên Hintz được tìm thấy.
<div>SQL&gt; CẬP NHẬT nhân viên</div> <div>Mức lương BỘ = 7000</div> <div>Ở ĐẦU Last_name='Banda';</div>	Không có hành động.	Phiên 1 bắt đầu giao dịch 1 bằng cách cập nhật mức lương Banda. Mức cách ly mặc định là READ TẬN TỤY.
Không có hành động.	<div>SQL&gt; THIẾT LẬP GIAO DỊCH</div> <div>MỨC ĐỘ CÔ LẬP CÓ THỂ TUẦN TỰ HÓA;</div>	Phiên 2 bắt đầu giao dịch 2 và đặt nó vào SERIALIZABLE mức độ cách ly.

Bảng 9-3 (Tiếp theo) Giao dịch có thể tuần tự hóa

Phiên 1	Buổi 2	Giải trình
Không có hành động.	<div>SQL&gt; CHỌN họ, lương TỪ nhân viên Ở ĐẦU họ_TRÊN ("Banda", "Greene", "Hintz");</div> <div><div>HỌLƯƠNG</div><div>Banda6200</div><div>Greene9500</div></div>	Giao dịch 2 truy vấn lương cho Banda, Greene và gợi ý. Sử dụng cơ sở dữ liệu Oracle đọc tính nhất quán để hiển thị lương cho Banda trước bản cập nhật không được cam kết được thực hiện bởi giao dịch 1.
Không có hành động.	<div>SQL&gt; CẬP NHẬT nhân viên Mức lương BỘ = 9900 Ở ĐẦU Last_name = 'Greene';</div>	Giao dịch 2 cập nhật Greene lương thành công bởi vì chỉ có hàng Banda là bị khóa.
<div>SQL&gt; XÁC NHẬN VÀO nhân viên (employee_id, họ_tên, email, ngày thuê, job_id) GIÁ TRỊ (210, 'Hintz', 'JHINTZ', SYSDATE, 'SH_CLERK');</div>	Không có hành động.	Giao dịch 1 chèn một hàng cho nhân viên Hintz.
SQL> CAM KẾT;	Không có hành động.	Giao dịch 1 cam kết công việc của nó, kết thúc giao dịch.
<div>SQL&gt; CHỌN họ, lương TỪ nhân viên Ở ĐẦU họ_TRÊN ("Banda", "Greene", "Hintz");</div> <div><div>HỌLƯƠNG</div><div>Banda7000</div><div>Greene9500</div></div> <div>gợi ý</div>	<div>SQL&gt; CHỌN họ, lương TỪ nhân viên Ở ĐẦU họ_TRÊN ("Banda", "Greene", "Hintz");</div> <div><div>HỌLƯƠNG</div><div>Banda6200</div><div>Greene9900</div></div>	<div>Phần 1 truy vấn mức lương của nhân viên Banda, Greene, và Hintz và nhìn thấy những thay đổi cam kết bởi giao dịch 1. Bài 1 không thấy bản cập nhật Greene không được cam kết được thực hiện bởi giao dịch 2.</div> <div>Giao dịch 2 truy vấn lương cho nhân viên Banda, Greene và Hintz. Lời tiên tri Tính nhất quán khi đọc cơ sở dữ liệu đảm bảo rằng phần chèn Hintz và Cập nhật Banda được cam kết bởi giao dịch 1 không hiển thị với giao dịch 2. Giao dịch 2 nhìn thấy bản cập nhật riêng của nó cho Greene Lương.</div>
Không có hành động.	LÀM;	Giao dịch 2 cam kết công việc của nó, kết thúc giao dịch.

Bảng 9-3 (Tiếp theo) Giao dịch có thể tuần tự hóa

Phiên 1	Buổi 2	Giải trình
<div>SQL&gt; CHỌN họ, lương</div> <div>TỪ nhân viên</div> <div>Ở ĐẦU họ_TRÊN</div> <div>("Banda", "Greene", "Hintz");</div> <div><div>HỌ</div><div>LƯƠNG</div><div><div>Banda7000</div><div>Greene9900</div></div></div> <div>gợi ý</div>	<div>SQL&gt; CHỌN họ, lương</div> <div>TỪ nhân viên</div> <div>Ở ĐẦU họ_TRÊN</div> <div>("Banda", "Greene", "Hintz");</div> <div><div>HỌ</div><div>LƯƠNG</div><div><div>Banda7000</div><div>Greene9900</div></div></div> <div>gợi ý</div>	<div>Cả hai phiên đều truy vấn mức lương</div> <div>cho Banda, Greene và Hintz.</div> <div>Mỗi phiên nhìn thấy tất cả cam kết</div> <div>những thay đổi được thực hiện bởi giao dịch 1</div> <div>và giao dịch 2.</div>
<div>SQL&gt; CẬP NHẬT nhân viên</div> <div>Mức lương BỘ = 7100</div> <div>Ở ĐẦU Last_name = 'Hintz';</div>	<div>Không có hành động.</div>	<div>Phiên 1 bắt đầu giao dịch 3</div> <div>bằng cách cập nhật mức lương Hintz. Các</div> <div>mức cô lập mặc định cho</div> <div>giao dịch 3 được ĐỌC CAM KẾT.</div>
<div>Không có hành động.</div>	<div>SQL&gt; THIẾT LẬP GIAO DỊCH</div> <div>MỨC ĐỘ CÔ LẬP CÓ THỂ TUẦN TIẾP;</div>	<div>Phiên 2 bắt đầu giao dịch 4</div> <div>và đặt nó vào SERIALIZABLE</div> <div>mức độ cách ly.</div>
<div>Không có hành động.</div>	<div>SQL&gt; CẬP NHẬT nhân viên</div> <div>Mức lương BỘ = 7200</div> <div>Ở ĐẦU Last_name = 'Hintz';</div> <div>-- lời nhắc không trả về</div>	<div>Giao dịch 4 lần thử cập nhật</div> <div>mức lương cho Hintz, nhưng</div> <div>bị chặn vì giao dịch 3</div> <div>đã khóa hàng Hintz. Giao dịch</div> <div>4 hàng đợi phía sau giao dịch 3.</div>
<div>SQL&gt; CAM KẾT;</div>	<div>Không có hành động.</div>	<div>Giao dịch 3 cam kết cập nhật</div> <div>của mức lương Hintz, kết thúc</div> <div>giao dịch.</div>
<div>Không có hành động.</div>	<div>CẬP NHẬT nhân viên</div> <div>Mức lương BỘ = 7200</div> <div>Ở ĐẦU họ_ = 'Hintz'</div> <div>*  LỖI ở dòng 1: ORA-08177: không thể tuần tự hóa quyền truy cập cho giao dịch này</div>	<div>Cam kết kết thúc giao dịch</div> <div>3 nguyên nhân gây ra bản cập nhật Hintz trong</div> <div>giao dịch 4 thất bại với</div> <div>LỖI ORA-08177 . Vấn đề</div> <div>xảy ra lỗi do giao dịch</div> <div>3 cam kết cập nhật Hintz</div> <div>sau khi giao dịch 4 bắt đầu.</div>
<div>Không có hành động.</div>	<div>SQL&gt; HOÀN LẠI;</div>	<div>Phiên 2 khôi phục giao dịch</div> <div>4, kết thúc giao dịch.</div>
<div>Không có hành động.</div>	<div>SQL&gt; THIẾT LẬP GIAO DỊCH</div> <div>MỨC ĐỘ CÔ LẬP CÓ THỂ TUẦN TIẾP;</div>	<div>Phiên 2 bắt đầu giao dịch 5</div> <div>và đặt nó vào SERIALIZABLE</div> <div>mức độ cách ly.</div>

Bảng 9-3 (Tiếp theo) Giao dịch có thể tuần tự hóa

Phiên 1	Buổi 2	Giải trình
Không có hành động.	<div>SQL&gt; CHON họ, lương TỪ nhân viên Ở ĐÂU họ_TRÊN ("Banda", "Greene", "Hintz");</div> <div><div>HỌLƯƠNG</div><div><div></div><div></div><div></div></div><div>Banda7000 Greene9500 gợi ý7100</div></div>	Giao dịch 5 truy vấn lương cho Banda, Greene và gợi ý. Cập nhật lương Hintz cam kết bởi giao dịch 3 là dễ thấy.
Không có hành động.	<div>SQL&gt; CẬP NHẬT nhân viên Mức lương BỘ = 7200 Ở ĐÂU Last_name='Hintz';</div> <div>Đã cập nhật 1 hàng.</div>	Giao dịch 5 cập nhật Hintz lương sang một giá trị khác. Bởi vì bản cập nhật Hintz đã thực hiện bởi giao dịch 3 đã cam kết trước khi bắt đầu giao dịch 5, vấn đề truy cập nổi tiếp là tránh được.  Lưu ý: Nếu giao dịch khác đã cập nhật và cam kết Hintz hàng sau khi giao dịch 5 bắt đầu, sau đó truy cập nổi tiếp vấn đề sẽ lại xảy ra.
Không có hành động.	SQL> CAM KẾT;	Phiên 2 cam kết cập nhật không có vấn đề gì, kết thúc giao dịch.



Xem thêm:

- "Khóa hàng (TX)"
- "Tổng quan về kiểm soát giao dịch"

## Mức cô lập chỉ đọc

Mức cô lập chỉ đọc tương tự như mức cô lập tuần tự hóa, nhưng chỉ đọc giao dịch không cho phép sửa đổi dữ liệu trong giao dịch trừ khi người dùng là SYS.

Các giao dịch chỉ đọc không dễ bị lỗi ORA-08177 . Chỉ đọc giao dịch rất hữu ích để tạo báo cáo trong đó nội dung phải nhất quán đối với thời điểm giao dịch bắt đầu.

Cơ sở dữ liệu Oracle đạt được tính nhất quán khi đọc bằng cách xây dựng lại dữ liệu khi cần thiết từ các phân đoạn hoàn tác. Bởi vì các phân đoạn hoàn tác được sử dụng theo kiểu vòng tròn, nên cơ sở dữ liệu có thể ghi đè dữ liệu hoàn tác. Các báo cáo chạy dài có nguy cơ hoàn tác dữ liệu cần thiết cho tính nhất quán khi đọc có thể đã được sử dụng lại bởi một giao dịch khác, làm tăng một lỗi chập nhanh quá cũ . Đặt khoảng thời gian lưu giữ hoàn tác, đây là thời gian tối thiểu khoảng thời gian cơ sở dữ liệu cố gắng giữ lại dữ liệu hoàn tác cũ trước khi ghi đè lên nó, tránh được vấn đề này một cách thích hợp.



Xem thêm:

- "Hoàn tác phân đoạn"
- Hướng dẫn dành cho quản trị viên cơ sở dữ liệu Oracle để tìm hiểu cách thiết lập hoàn tác thời gian lưu giữ

## Tổng quan về Cơ chế khóa cơ sở dữ liệu Oracle

Khóa là một cơ chế ngăn chặn các tương tác phá hoại .

Các tương tác có tính hủy diệt khi chúng cập nhật dữ liệu không chính xác hoặc thay đổi không chính xác cấu trúc dữ liệu cơ bản giữa các giao dịch truy cập dữ liệu được chia sẻ. Khóa đóng một vai trò quan trọng trong việc duy trì tính đồng thời và nhất quán của cơ sở dữ liệu.

## Tóm tắt hành vi khóa

Cơ sở dữ liệu duy trì một số loại khóa khác nhau, tùy thuộc vào hoạt động thu được khóa.

Nói chung, cơ sở dữ liệu sử dụng hai loại khóa: khóa độc quyền và khóa chia sẻ.

Chỉ có thể lấy được một khóa độc quyền trên một tài nguyên như một hàng hoặc một bảng, nhưng có thể lấy được nhiều khóa chia sẻ trên một tài nguyên.

Khóa ảnh hưởng đến sự tương tác của người đọc và người viết. Trình đọc là một truy vấn về tài nguyên, trong khi trình soạn thảo là một câu lệnh sửa đổi tài nguyên. Các quy tắc sau đây tóm tắt hành vi khóa của Cơ sở dữ liệu Oracle đối với người đọc và người viết:

- Một hàng chỉ bị khóa khi được người viết sửa đổi.

Khi một câu lệnh cập nhật một hàng, giao dịch sẽ chỉ lấy một khóa cho hàng này. Bằng cách khóa dữ liệu bảng ở cấp hàng, cơ sở dữ liệu sẽ giảm thiểu sự tranh chấp đối với cùng một dữ liệu. Trong các trường hợp bình thường<sup>1</sup> cơ sở dữ liệu không chuyển khóa hàng lên cấp khối hoặc bảng.

- Người viết của một hàng sẽ chặn người viết cùng một hàng.

Nếu một giao dịch đang sửa đổi một hàng thì khóa hàng sẽ ngăn giao dịch khác sửa đổi cùng một hàng cùng một lúc.

- Người đọc không bao giờ chặn người viết.

Bởi vì người đọc hàng không khóa nó nên người viết có thể sửa đổi hàng này. Ngoại lệ duy nhất là câu lệnh SELECT ... FOR UPDATE , đây là một loại câu lệnh SELECT đặc biệt có chức năng khóa hàng mà nó đang đọc.

- Nhà văn không bao giờ chặn người đọc.

Khi một hàng đang được người viết thay đổi, cơ sở dữ liệu sẽ sử dụng dữ liệu hoàn tác để cung cấp cho người đọc cái nhìn nhất quán về hàng đó.

<sup>1</sup> Khi xử lý cam kết hai pha phân tán, cơ sở dữ liệu có thể tạm thời ngăn chặn quyền truy cập đọc trong các trường hợp đặc biệt. Cụ thể, nếu một truy vấn bắt đầu giữa các giai đoạn chuẩn bị và chuyển giao và cố gắng đọc dữ liệu trước khi chuyển giao thì cơ sở dữ liệu có thể chuyển khóa từ cấp hàng lên cấp khối để đảm bảo tính nhất quán khi đọc.



**Ghi chú:**

Người đọc dữ liệu có thể phải đợi người ghi cùng khối dữ liệu trong những trường hợp rất đặc biệt về các giao dịch phân tán đang chờ xử lý.

**Xem thêm:**

- Tham khảo ngôn ngữ SQL của cơ sở dữ liệu Oracle để tìm hiểu về SELECT ...  
ĐỂ CẬP NHẬT
- Hướng dẫn dành cho quản trị viên cơ sở dữ liệu Oracle để tìm hiểu về các thời gian chờ liên quan với các giao dịch phân tán đáng nghi ngờ

## Sử dụng ổ khóa

Trong cơ sở dữ liệu một người dùng, khóa không cần thiết vì chỉ có một người dùng sửa đổi thông tin. Tuy nhiên, khi nhiều người dùng truy cập và sửa đổi dữ liệu, cơ sở dữ liệu phải cung cấp cách ngăn chặn việc sửa đổi đồng thời cùng một dữ liệu.

Khóa đạt được các yêu cầu cơ sở dữ liệu quan trọng sau:

- Tính nhất quán Dữ

liệu mà một phiên đang xem hoặc thay đổi không được thay đổi bởi các phiên khác cho đến khi người dùng kết thúc.

- Chính trực

Dữ liệu và cấu trúc phải phản ánh tất cả các thay đổi được thực hiện theo đúng trình tự.

Cơ sở dữ liệu Oracle cung cấp tính đồng thời, tính nhất quán và tính toàn vẹn của dữ liệu giữa các giao dịch thông qua cơ chế khóa của nó. Việc khóa diễn ra tự động và không yêu cầu người dùng thực hiện hành động nào.

Sự cần thiết của khóa có thể được minh họa bằng việc cập nhật đồng thời một hàng. Trong ví dụ sau, một ứng dụng dựa trên web đơn giản sẽ cung cấp cho người dùng cuối email và số điện thoại của nhân viên.

Ứng dụng sử dụng câu lệnh UPDATE như sau để sửa đổi dữ liệu:

**CẬP NHẬT nhân viên**

BỘ email = ?, số điện thoại = ?

Nhân viên\_id = ở đâu? email = ?

VÀ

VÀ số\_điện thoại = ?

Trong câu lệnh UPDATE trước đó, các giá trị email và số điện thoại trong mệnh đề WHERE là các giá trị ban đầu, chưa sửa đổi cho nhân viên được chỉ định. Bản cập nhật này đảm bảo rằng hàng mà ứng dụng sửa đổi không bị thay đổi sau lần cuối ứng dụng đọc và hiển thị nó cho người dùng. Bằng cách này, ứng dụng sẽ tránh được vấn đề mất bản cập nhật trong đó một người dùng ghi đè các thay đổi do người dùng khác thực hiện, khiến người dùng thứ hai mất bản cập nhật (Bảng 9-2 cho thấy một ví dụ về bản cập nhật bị mất).

Bảng 9-4 Ví dụ về khóa hàng

17-9

Bảng 9-4 (Tiếp) Ví dụ về khóa hàng

T Phiên 1	Buổi 2	Sự miêu tả
t6	<p>CẬP NHẬT nhân viên SET  phone_number='515.555.1235'  Ở ĐẦU nhân_viên_id=118  VÀ email='GHIMURO' VÀ  phone_number='515.555.1234';</p> <p>Đã cập nhật 1 hàng.</p>	<p>Trong phiên 1, người dùng hr1 nhận ra rằng họ đã cập nhật hàng GHIMURO sai số điện thoại. Người dùng bắt đầu một giao dịch mới và cập nhật số điện thoại trong hàng thành 515.555.1235, khóa hàng GHIMURO .</p>
t7	<p>CHỌN nhân_viên_id làm ID, email, số điện thoại  TỪ hr.employees Ở ĐẦU  Last_name='Himuro';</p> <p>ID EMAIL PHONE_NUMBER  -----  118 GHIMURO 515.555.1234</p>	<p>Trong phiên 2, người dùng hr2 truy vấn hr.employees để tìm bản ghi Himuro. Bản ghi hiển thị cập nhật số điện thoại được cam kết bởi phiên 1 lúc t4. Tính nhất quán trong việc đọc Cơ sở dữ liệu Oracle đảm bảo rằng phiên 2 không thấy thay đổi không được cam kết được thực hiện tại t6.</p>
t8	<p>CẬP NHẬT hr.employees SET  phone_number='515.555.1235 WHERE nhân_viên_id=118 VÀ email='GHIMURO'</p> <p>VÀ phone_number =  '515.555.1234';</p> <p>-- SQL*Plus không hiển thị -- một thông báo cập nhật hàng hoặc -- trả về lời nhắc.</p>	<p>Trong phiên 2, người dùng hr2 cố gắng cập nhật cùng một hàng nhưng bị chặn vì hr1 hiện đang xử lý hàng ngang.</p>
t9	<p>HOÀN LẠI;</p> <p>Hoàn tất khôi phục.</p>	<p>Trong phiên 1, người dùng hr1 khôi phục giao dịch và kết thúc giao dịch.</p>
t10	<p>Đã cập nhật 1 hàng.</p>	<p>Trong phiên 2, việc cập nhật số điện thoại thành công vì bản cập nhật phiên 1 đã được khôi phục. Hàng GHIMURO khớp với vị tử của nó nên quá trình cập nhật thành công.</p>
t11	<p>LÀM;</p> <p>Cam kết hoàn tất.</p>	<p>Phiên 2 cam kết cập nhật, kết thúc giao dịch.</p>

Cơ sở dữ liệu Oracle tự động lấy các khóa cần thiết khi thực thi các câu lệnh SQL.

Ví dụ: trước khi cơ sở dữ liệu cho phép một phiên sửa đổi dữ liệu, trước tiên phiên đó phải khóa dữ liệu. Khóa cung cấp cho phiên quyền kiểm soát dữ liệu độc quyền để không giao dịch nào khác có thể sửa đổi dữ liệu bị khóa cho đến khi khóa được giải phóng.

Do cơ chế khóa của Cơ sở dữ liệu Oracle gắn chặt với kiểm soát giao dịch nên người thiết kế ứng dụng chỉ cần xác định đúng các giao dịch và Cơ sở dữ liệu Oracle sẽ tự động quản lý việc khóa. Người dùng không bao giờ cần khóa bất kỳ tài nguyên nào một cách rõ ràng, mặc dù Cơ sở dữ liệu Oracle cũng cho phép người dùng khóa dữ liệu theo cách thủ công.

Các phần sau đây giải thích các khái niệm quan trọng để hiểu cách Cơ sở dữ liệu Oracle đạt được tính đồng thời của dữ liệu.



### Xem thêm:

Tham khảo các loại và gói PL/SQL của Cơ sở dữ liệu Oracle để tìm hiểu về Gói OWA\_OPT\_LOCK , chứa các chương trình con có thể giúp ngăn chặn các bản cập nhật bị mất

## Chế độ khóa

Cơ sở dữ liệu Oracle tự động sử dụng mức hạn chế có thể áp dụng thấp nhất để cung cấp mức độ đồng thời dữ liệu cao nhất nhưng vẫn đảm bảo tính toàn vẹn dữ liệu an toàn.

Mức độ càng ít hạn chế thì dữ liệu càng có sẵn để người dùng khác truy cập.

Ngược lại, mức độ càng hạn chế thì các giao dịch khác về loại khóa mà họ có thể có được càng bị hạn chế.

Cơ sở dữ liệu Oracle sử dụng hai chế độ khóa trong cơ sở dữ liệu nhiều người dùng:

- Chế độ khóa độc quyền

Chế độ này ngăn chặn việc chia sẻ tài nguyên liên quan. Một giao dịch nhận được một **khóa độc quyền** khi nó sửa đổi dữ liệu. Giao dịch đầu tiên khóa riêng một tài nguyên là giao dịch duy nhất có thể thay đổi tài nguyên cho đến khi khóa độc quyền được giải phóng.

- Chế độ khóa chia sẻ

Chế độ này cho phép chia sẻ tài nguyên liên quan, tùy thuộc vào các hoạt động liên quan. Nhiều người dùng đọc dữ liệu có thể chia sẻ dữ liệu, mỗi người giữ một **khóa chia sẻ** để ngăn người viết cần khóa độc quyền truy cập đồng thời. Nhiều giao dịch có thể có được khóa chia sẻ trên cùng một tài nguyên.

Giả sử rằng một giao dịch sử dụng câu lệnh SELECT ... FOR UPDATE để chọn một hàng trong bảng. Giao dịch có được khóa hàng độc quyền và khóa bảng chia sẻ hàng.

Khóa hàng cho phép các phiên khác sửa đổi bất kỳ hàng nào ngoài hàng bị khóa, trong khi khóa bảng ngăn các phiên thay đổi cấu trúc của bảng. Do đó, cơ sở dữ liệu cho phép thực thi càng nhiều câu lệnh càng tốt.

## Khóa chuyển đổi và nâng cấp

Cơ sở dữ liệu Oracle thực hiện chuyển đổi khóa khi cần thiết.

Trong quá trình chuyển đổi khóa, cơ sở dữ liệu sẽ tự động chuyển đổi khóa bảng có mức hạn chế thấp hơn thành khóa có mức hạn chế cao hơn. Ví dụ: giả sử một giao dịch đưa ra lệnh SELECT ... FOR UPDATE cho một nhân viên và sau đó cập nhật hàng bị khóa. Trong trường hợp này, cơ sở dữ liệu sẽ tự động chuyển đổi khóa bảng chia sẻ hàng thành khóa bảng dành riêng cho hàng. Một giao dịch giữ các khóa hàng độc quyền cho tất cả các hàng được chèn, cập nhật hoặc xóa trong giao dịch. Vì khóa hàng được thu thập ở mức độ hạn chế cao nhất nên không cần thực hiện hoặc yêu cầu chuyển đổi khóa.

Chuyển đổi khóa khác với **leo thang khóa**, xảy ra khi nhiều khóa được giữ ở một mức độ chi tiết (ví dụ: hàng) và cơ sở dữ liệu nâng khóa lên mức độ chi tiết cao hơn (ví dụ: bảng). Nếu một phiên khóa nhiều hàng trong một

bảng, sau đó một số cơ sở dữ liệu sẽ tự động chuyển khóa hàng thành một bảng duy nhất. Số lượng ổ khóa giảm đi nhưng mức độ hạn chế của những gì bị khóa lại tăng lên.

Cơ sở dữ liệu Oracle không bao giờ leo thang khóa. Việc leo thang khóa làm tăng đáng kể khả năng xảy ra bế tắc. Giả sử rằng một hệ thống đang cố gắng nâng cấp khóa thay mặt cho giao dịch 1 nhưng không thể do khóa được giữ bởi giao dịch 2. Gián đoạn được tạo ra nếu giao dịch 2 cũng yêu cầu leo thang khóa của cùng một dữ liệu trước khi có thể tiếp tục.

## Thời lượng khóa

Cơ sở dữ liệu Oracle tự động mở khóa khi một số sự kiện xảy ra để giao dịch không còn yêu cầu tài nguyên nữa.

Thông thường, cơ sở dữ liệu lưu giữ các khóa có được bằng các câu lệnh trong một giao dịch trong suốt thời gian giao dịch. Các khóa này ngăn chặn sự can thiệp mang tính hủy diệt như đọc sai, mất cập nhật và [DDL](#) phá hoại từ các giao dịch đồng thời.



### Ghi chú:

Khóa bảng được lấy trên bảng con do khóa ngoại không được lập chỉ mục sẽ được giữ trong suốt thời gian của câu lệnh chứ không phải giao dịch. Ngoài ra, gói DBMS\_LOCK cho phép các khóa do người dùng xác định được giải phóng và phân bổ theo ý muốn và thậm chí được giữ trên ranh giới giao dịch.

Cơ sở dữ liệu Oracle giải phóng tất cả các khóa có được bởi các câu lệnh trong một giao dịch khi nó cam kết hoặc khôi phục. Cơ sở dữ liệu Oracle cũng giải phóng các khóa có được sau một [điểm lưu trữ](#) khi quay trở lại điểm lưu trữ. Tuy nhiên, chỉ những giao dịch không chờ đợi các tài nguyên đã bị khóa trước đó mới có thể có được các khóa trên các tài nguyên hiện có. Các giao dịch chờ đợi tiếp tục đợi cho đến khi giao dịch ban đầu được cam kết hoặc quay trở lại hoàn toàn.



### Xem thêm:

- "[Bảng 10-3](#)" thể hiện hành vi chờ giao dịch
- "[Tổng quan về Khóa do người dùng xác định](#)" để tìm hiểu thêm về DBMS\_LOCK

## Khóa và bế tắc

Bế tắc là tình huống trong đó có hai hoặc nhiều người dùng đang chờ dữ liệu bị khóa lẫn nhau. Sự bế tắc ngăn cản một số giao dịch tiếp tục hoạt động.

Cơ sở dữ liệu Oracle tự động phát hiện các bế tắc và giải quyết chúng bằng cách khôi phục một câu lệnh liên quan đến bế tắc, giải phóng một bộ khóa hàng xung đột.

Cơ sở dữ liệu trả về một thông báo tương ứng cho giao dịch trải qua [quá trình khôi phục ở cấp độ câu lệnh](#). Câu lệnh được khôi phục thuộc về giao dịch phát hiện bế tắc. Thông thường, giao dịch được báo hiệu phải được khôi phục một cách rõ ràng, nhưng nó có thể thử lại câu lệnh khôi phục sau khi chờ đợi.

Bảng 9-5 minh họa hai giao dịch trong tình trạng bế tắc.

Bảng 9-5 Giao dịch bế tắc

T Phiên 1	Buổi 2	Giải trình
t0  SQL> CẬP NHẬT nhân viên SET lương = lương*1.1 WHERE nhân viên_id = 100;  Đã cập nhật 1 hàng.	SQL> CẬP NHẬT nhân viên SET lương = lương*1.1 WHERE nhân viên_id = 200;  Đã cập nhật 1 hàng.	Phiên 1 bắt đầu giao dịch 1 và cập nhật mức lương cho nhân viên 100. Phiên 2 bắt đầu giao dịch 2 và cập nhật mức lương cho nhân viên 200. Không có vấn đề gì tồn tại vì mỗi giao dịch chỉ khóa hàng mà nó cố cập nhật.
t1  SQL> CẬP NHẬT nhân viên SET lương = lương*1.1 WHERE nhân viên_id = 200;  -- lời nhắc không trả về	SQL> CẬP NHẬT nhân viên SET lương = lương*1.1 WHERE nhân viên_id = 100;  -- lời nhắc không trả về	Giao dịch 1 cố gắng cập nhật hàng 200 của nhân viên, hiện đang bị khóa bởi giao dịch 2. Giao dịch 2 cố gắng cập nhật hàng 100 nhân viên, hiện đang bị khóa bởi giao dịch 1. Kết quả là bế tắc vì không giao dịch nào có thể lấy được tài nguyên cần thiết để tiếp tục hoặc kết thúc. Cho dù mỗi giao dịch có chờ bao lâu thì các khóa xung đột vẫn được giữ.
t2  CẬP NHẬT nhân viên *  Lỗi ở dòng 1: ORA-00060: phát hiện bế tắc trong khi chờ tài nguyên  SQL>		Giao dịch 1 báo hiệu sự bế tắc và khôi phục câu lệnh UPDATE được ban hành tại t1. Tuy nhiên, bản cập nhật được thực hiện vào thời điểm t0 không được khôi phục. Lời nhắc được trả về trong phiên 1. Lưu ý: Chỉ một phiên trong tình trạng bế tắc thực sự gặp lỗi bế tắc, nhưng cả hai phiên đều có thể gặp lỗi.
t3  SQL> CAM KẾT;  Cam kết hoàn tất.		Phiên 1 cam kết cập nhật được thực hiện tại t0, kết thúc giao dịch 1. Cập nhật không thành công tại thời điểm t1 không được cam kết.
t4	Đã cập nhật 1 hàng.  SQL>	Cập nhật tại t1 trong giao dịch 2, vốn đang bị chặn bởi giao dịch 1, được thực thi. Lời nhắc được trả về.
t5	SQL> CAM KẾT;  Cam kết hoàn tất.	Phiên 2 cam kết các cập nhật được thực hiện tại t0 và t1, kết thúc giao dịch 2.

Bế tắc thường xảy ra nhất khi các giao dịch ghi đè rõ ràng khóa mặc định của Cơ sở dữ liệu Oracle. Bởi vì Cơ sở dữ liệu Oracle không chuyển cấp khóa và không sử dụng khóa đọc cho các truy vấn, nhưng sử dụng khóa cấp hàng (chứ không phải cấp trang), nên bế tắc hiếm khi xảy ra.

**Xem thêm:**

- ["Tổng quan về khóa dữ liệu thủ công"](#)
- Hướng dẫn phát triển cơ sở dữ liệu Oracle để tìm hiểu cách xử lý bế tắc khi bạn khóa bảng một cách rõ ràng

## Tổng quan về khóa tự động

Cơ sở dữ liệu Oracle tự động khóa tài nguyên thay mặt cho một giao dịch để ngăn các giao dịch khác thực hiện điều gì đó yêu cầu quyền truy cập độc quyền vào cùng một tài nguyên nguồn.

Cơ sở dữ liệu tự động thu được các loại khóa khác nhau ở các mức hạn chế khác nhau tùy thuộc vào tài nguyên và thao tác đang được thực hiện.

**Ghi chú:**

Cơ sở dữ liệu không bao giờ khóa các hàng khi thực hiện các thao tác đọc đơn giản.

Khóa cơ sở dữ liệu Oracle được chia thành các loại hiển thị trong bảng sau.

Bảng 9-6 Danh mục khóa

Kho a	Sự miêu tả	Để tìm hiểu thêm
Khóa DML	Bảo vệ dữ liệu. Ví dụ: khóa bảng khóa toàn bộ bảng, trong khi khóa hàng khóa các hàng đã chọn.	<a href="#">"Khóa DML"</a>
Khóa DDL	Bảo vệ cấu trúc của các đối tượng lược đồ— ví dụ: định nghĩa từ điển của bảng và dạng xem.	<a href="#">"Khóa DDL"</a>
Khóa hệ thống	Bảo vệ cấu trúc cơ sở dữ liệu nội bộ như dưới dạng các tập tin dữ liệu. Chốt, mutexes và khóa bên trong hoàn toàn tự động.	<a href="#">"Khóa hệ thống"</a>

## Khóa DML

Khóa DML, còn được gọi là khóa dữ liệu, đảm bảo tính toàn vẹn của dữ liệu được nhiều người dùng truy cập đồng thời.

Ví dụ: khóa DML ngăn hai khách hàng mua bản sao cuối cùng của cuốn sách có sẵn từ người bán sách trực tuyến. Khóa DML ngăn chặn sự can thiệp mang tính hủy diệt của các hoạt động DML hoặc DDL xung đột đồng thời.

Câu lệnh DML tự động thu được các loại khóa sau:

- [Khóa hàng \(TX\)](#)
- [Khóa bản \(TM\)](#)

Trong các phần sau, từ viết tắt trong ngoặc đơn sau mỗi loại khóa hoặc chế độ khóa là từ viết tắt được sử dụng trong Locks Monitor của Oracle Enterprise Manager (Enterprise Manager). Trình quản lý doanh nghiệp có thể hiển thị TM cho bất kỳ khóa bảng nào thay vì chỉ ra chế độ khóa bảng (chẳng hạn như RS hoặc SRX).



Xem thêm:

"Trình quản lý doanh nghiệp Oracle"

## Khóa hàng (TX)

Khóa hàng, còn được gọi là khóa TX, là khóa trên một hàng của bảng. Một giao dịch có được một khóa hàng cho mỗi hàng được sửa đổi bằng câu lệnh INSERT, UPDATE, DELETE, MERGE hoặc SELECT ... FOR UPDATE. Khóa hàng tồn tại cho đến khi giao dịch được thực hiện hoặc quay trở lại.

Khóa hàng chủ yếu đóng vai trò như một cơ chế xếp hàng để ngăn hai giao dịch sửa đổi cùng một hàng. Cơ sở dữ liệu luôn khóa một hàng đã sửa đổi ở chế độ độc quyền để các giao dịch khác không thể sửa đổi hàng đó cho đến khi giao dịch giữ khóa được xác nhận hoặc quay lại. Khóa hàng cung cấp khả năng khóa hạt tốt nhất có thể và do đó mang lại khả năng xử lý đồng thời và thông lượng tốt nhất có thể.



Ghi chú:

Nếu một giao dịch chấm dứt do **lỗi phiên bản** cơ sở dữ liệu thì quá trình khôi phục ở cấp khối sẽ cung cấp một hàng trước khi toàn bộ giao dịch được khôi phục.

Nếu một giao dịch nhận được khóa cho một hàng thì giao dịch đó cũng nhận được khóa cho bảng chứa hàng đó. Khóa bảng ngăn chặn các hoạt động DDL xung đột có thể ghi đè các thay đổi dữ liệu trong giao dịch hiện tại. **Hình 9-2** minh họa việc cập nhật hàng thứ ba trong bảng. Cơ sở dữ liệu Oracle tự động đặt một khóa độc quyền trên hàng được cập nhật và một khóa độc quyền phụ trên bảng.



Hình 9-2 Khóa hàng và khóa bảng

Bảng NHÂN VIÊN

MÃ HIỆU CÔNG NHÂN	LAST_NAME	EMAIL	HIRE_DATE	CÔNG VIỆC_ID	QUẢN LÝ_ID BỘ	PHÂN_ID
100	Nhà vua	TRUỘT TRUỘT	17-06-87	AD_PRES		90
101	Kochhar	NKOCHAR	21-09-89	AD_VP	100	90
102	De Hann	LDEHANN	13-01-93	AD_VP	100	90
103	Hunold	AHUNOLD	03-01-90	IT_PROG	102	60

Đã có được khóa bảng

Đã có được khóa hàng độc quyền (TX)

Hàng đang được cập nhật

Khóa hàng và đồng thời

Kịch bản này minh họa cách Cơ sở dữ liệu Oracle sử dụng khóa hàng cho hoạt động đồng thời.

Ba phiên truy vấn cùng một hàng cùng một lúc. Phiên 1 và 2 tiến hành thực hiện các cập nhật có sẵn cho các hàng khác nhau, trong khi phiên 3 không thực hiện cập nhật. Mỗi phiên nhìn thấy các bản cập nhật không được cam kết của riêng mình nhưng không nhìn thấy các bản cập nhật không được cam kết của bất kỳ phiên nào khác.

Bảng 9-7 Ví dụ về đồng thời dữ liệu

T	Phiên 1	Buổi 2	Buổi 3	Giải trình
t0	<div>CHỌN nhân viên_id, mức lương TỪ nhân viên Ở ĐẦU nhân viên_id TRONG ( 100, 101 );</div> <div>LƯƠNG NHÂN VIÊN_ID</div> <div><div>100512</div><div>101600</div></div>	<div>CHỌN nhân viên_id, mức lương TỪ nhân viên Ở ĐẦU nhân viên_id TRONG ( 100, 101 );</div> <div>LƯƠNG NHÂN VIÊN_ID</div> <div><div>100512</div><div>101600</div></div>	<div>CHỌN nhân viên_id, mức lương TỪ nhân viên Ở ĐẦU nhân viên_id TRONG ( 100, 101 );</div> <div>LƯƠNG NHÂN VIÊN_ID</div> <div><div>100512</div><div>101600</div></div>	Ba phiên khác nhau truy vấn đồng thời ID và lương của nhân viên 100 và 101. Kết quả trả về của mỗi truy vấn là giống hệt nhau.
t1	<div>CẬP NHẬT nhân viên BỘ lương = lương+100</div> <div>ở ĐẦU nhân viên_id=100;</div>			Đợt 1 cập nhật lương của nhân viên 100 nhưng không cam kết. Trong bản cập nhật, người viết chỉ lấy được khóa cấp hàng cho hàng được cập nhật, do đó ngăn người viết khác sửa đổi hàng này.

T	Phiên 1	Buổi 2	Buổi 3	Giải trình												
t2	<p>CHỌN nhân viên_id, lương</p> <p>TỪ nhân viên</p> <p>Ở ĐẦU nhân viên_id TRONG ( 100, 101 );</p> <p>LƯƠNG NHÂN VIÊN_ID</p> <hr/> <table> <tr> <td>100</td> <td>612</td> </tr> <tr> <td>101</td> <td>600</td> </tr> </table>	100	612	101	600	<p>CHỌN nhân viên_id, lương</p> <p>TỪ nhân viên</p> <p>Ở ĐẦU nhân viên_id TRONG ( 100, 101 );</p> <p>LƯƠNG NHÂN VIÊN_ID</p> <hr/> <table> <tr> <td>100</td> <td>512</td> </tr> <tr> <td>101</td> <td>600</td> </tr> </table>	100	512	101	600	<p>CHỌN nhân viên_id, mức lương</p> <p>TỪ nhân viên</p> <p>Ở ĐẦU nhân viên_id TRONG ( 100, 101 );</p> <p>LƯƠNG NHÂN VIÊN_ID</p> <hr/> <table> <tr> <td>100</td> <td>512</td> </tr> <tr> <td>101</td> <td>600</td> </tr> </table>	100	512	101	600	<p>Mỗi buổi</p> <p>vấn đề đồng thời truy vấn ban đầu. Phần 1 cho thấy kết quả lương 612 từ bản cập nhật t1. Các độc giả trong phần 2 và 3 hàng trả về ngay lập tức và không chờ tập 1 đi</p> <p>kết thúc giao dịch của nó. Các sử dụng cơ sở dữ liệu đọc đa phiên bản tính nhất quán để thể hiện mức lương như đã tồn tại trước khi cập nhật vào phiên 1.</p>
100	612															
101	600															
100	512															
101	600															
100	512															
101	600															
t3		<p>CẬP NHẬT nhân viên</p> <p>BỘ mức lương = lương +100</p> <p>ở đầu</p> <p>nhân viên_id=101;</p>		<p>Phiên 2 cập nhật lương nhân viên 101, nhưng không cam kết sự giao dịch. bên trong cập nhật, người viết đạt được cấp hàng khóa để cập nhật chỉ hàng, ngăn chặn các nhà văn khác từ sửa đổi hàng này.</p>												
t4	<p>CHỌN nhân viên_id, lương</p> <p>TỪ nhân viên</p> <p>Ở ĐẦU nhân viên_id TRONG ( 100, 101 );</p> <p>LƯƠNG NHÂN VIÊN_ID</p> <hr/> <table> <tr> <td>100</td> <td>612</td> </tr> <tr> <td>101</td> <td>600</td> </tr> </table>	100	612	101	600	<p>CHỌN nhân viên_id, lương</p> <p>TỪ nhân viên</p> <p>Ở ĐẦU nhân viên_id TRONG ( 100, 101 );</p> <p>LƯƠNG NHÂN VIÊN_ID</p> <hr/> <table> <tr> <td>100</td> <td>512</td> </tr> <tr> <td>101</td> <td>700</td> </tr> </table>	100	512	101	700	<p>CHỌN nhân viên_id, mức lương</p> <p>TỪ nhân viên</p> <p>Ở ĐẦU nhân viên_id TRONG ( 100, 101 );</p> <p>LƯƠNG NHÂN VIÊN_ID</p> <hr/> <table> <tr> <td>100</td> <td>512</td> </tr> <tr> <td>101</td> <td>600</td> </tr> </table>	100	512	101	600	<p>Mỗi buổi</p> <p>vấn đề đồng thời truy vấn ban đầu. Phần 1 cho thấy kết quả lương 612 từ bản cập nhật t1, nhưng không phải cập nhật Lương cho nhân viên 101 được thực hiện trong phiên 2.</p> <p>bạn đọc ở phần 2</p> <p>hiển thị mức lương cập nhật được thực hiện trong phiên 2, nhưng không phải lương cập nhật được thực hiện trong phiên 1. Người đọc trong phiên 3 sử dụng đọc tính nhất quán để thể hiện tiền lương trước đây</p> <p>sửa đổi theo phiên 1 và 2.</p>
100	612															
101	600															
100	512															
101	700															
100	512															
101	600															



### Xem thêm:

- Tham khảo ngôn ngữ SQL của cơ sở dữ liệu Oracle
- Tham khảo cơ sở dữ liệu Oracle để tìm hiểu về V\$LOCK

## Lưu trữ khóa hàng

Không giống như một số cơ sở dữ liệu sử dụng trình quản lý khóa để duy trì danh sách các khóa trong bộ nhớ, Cơ sở dữ liệu Oracle lưu trữ thông tin khóa trong khối dữ liệu chứa hàng bị khóa.

Cơ sở dữ liệu sử dụng cơ chế xếp hàng để thu được các khóa hàng. Nếu một giao dịch yêu cầu khóa cho một hàng đã được mở khóa thì giao dịch đó sẽ đặt một khóa vào khối dữ liệu. Mỗi hàng được giao dịch này sửa đổi sẽ trở về một bản sao ID giao dịch được lưu trong **tiêu đề khối**.

Khi giao dịch kết thúc, ID giao dịch vẫn còn trong tiêu đề khối. Nếu một giao dịch khác muốn sửa đổi một hàng thì nó sẽ sử dụng ID giao dịch để xác định xem khóa có hoạt động hay không. Nếu khóa đang hoạt động thì phiên sẽ yêu cầu được thông báo khi khóa được mở. Nếu không, giao dịch sẽ nhận được khóa.



### Xem thêm:

- "**Tổng quan về Khối dữ liệu**" để tìm hiểu thêm về tiêu đề khối dữ liệu
- Tham khảo cơ sở dữ liệu Oracle để tìm hiểu về V\$TRANSACTION

## Khóa bàn (TM)

Khóa bảng, còn được gọi là khóa TM, được giao dịch thu được khi bảng được sửa đổi bởi câu lệnh INSERT, UPDATE, DELETE, MERGE, SELECT với mệnh đề FOR UPDATE hoặc câu lệnh LOCK TABLE .

Các hoạt động DML yêu cầu khóa bảng để dành quyền truy cập DML vào bảng thay mặt cho một giao dịch và để ngăn các hoạt động DDL có thể xung đột với giao dịch.

Khóa bảng có thể được giữ ở bất kỳ chế độ nào sau đây:

- Chia sẻ hàng (RS)

Khóa này, còn được gọi là khóa bảng chia sẻ con (SS), cho biết rằng giao dịch giữ khóa trên bảng đã khóa các hàng trong bảng và có ý định cập nhật chúng. Khóa chia sẻ hàng là chế độ khóa bảng ít hạn chế nhất, cung cấp mức độ đồng thời cao nhất cho một bảng.

- Khóa bảng dành riêng cho hàng (RX)

Khóa này, còn được gọi là khóa bảng phụ (SX), thường chỉ ra rằng giao dịch giữ khóa đã cập nhật các hàng trong bảng hoặc đưa ra CHỌN ... ĐỂ CẬP NHẬT. Khóa SX cho phép các giao dịch khác truy vấn, chèn, cập nhật, xóa hoặc khóa các hàng đồng thời trong cùng một bảng. Vì vậy, khóa SX cho phép nhiều

giao dịch để có được đồng thời khóa SX và khóa bảng chia sẻ con cho cùng một bảng.

- Khóa bảng chia sẻ (S)

Khóa bảng chia sẻ do một giao dịch nắm giữ cho phép các giao dịch khác truy vấn bảng (không sử dụng CHỌN ... ĐỂ CẬP NHẬT), nhưng chỉ được phép cập nhật nếu một giao dịch duy nhất giữ khóa bảng chia sẻ. Bởi vì nhiều giao dịch có thể giữ một khóa bảng chia sẻ đồng thời, việc giữ khóa này là không đủ để đảm bảo rằng một giao dịch có thể sửa đổi bảng.

- Khóa bảng dành riêng cho hàng chia sẻ (SRX)

Khóa này, còn được gọi là khóa bảng chia sẻ độc quyền (SSX), hạn chế hơn khóa bảng chia sẻ. Mỗi lần chỉ có một giao dịch có thể có được khóa SSX trên một bảng nhất định. Khóa SSX được giữ bởi một giao dịch cho phép các giao dịch khác truy vấn bảng (ngoại trừ CHỌN ... ĐỂ CẬP NHẬT) nhưng không cập nhật bảng.

- Khóa bàn đọc quyền (X)

Khóa này là hạn chế nhất, cấm các giao dịch khác thực hiện bất kỳ loại câu lệnh DML nào hoặc đặt bất kỳ loại khóa nào trên bàn.



#### Xem thêm:

- Tham khảo ngôn ngữ SQL của cơ sở dữ liệu Oracle
- Hướng dẫn phát triển cơ sở dữ liệu Oracle để tìm hiểu thêm về khóa bảng

## Ổ khóa và chìa khóa ngoại

Cơ sở dữ liệu Oracle tối đa hóa khả năng kiểm soát đồng thời của các khóa gốc liên quan đến các khóa ngoại phụ thuộc.

Hành vi khóa phụ thuộc vào việc các cột khóa ngoại có được lập chỉ mục hay không. Nếu khóa ngoại không được lập chỉ mục thì bảng con có thể sẽ bị khóa thường xuyên hơn, bế tắc sẽ xảy ra và khả năng xử lý đồng thời sẽ giảm. Vì lý do này, khóa ngoại hầu như luôn được lập chỉ mục. Ngoại lệ duy nhất là khi khóa chính hoặc khóa duy nhất phù hợp không bao giờ được cập nhật hoặc xóa.

### Khóa và khóa ngoại chưa được lập chỉ mục

Cơ sở dữ liệu lấy được khóa bảng đầy đủ trên bảng con khi không có chỉ mục nào tồn tại trên cột khóa ngoại của bảng con và một phiên sửa đổi khóa chính trong bảng cha (ví dụ: xóa một hàng hoặc sửa đổi các thuộc tính khóa chính) hoặc hợp nhất các hàng vào bảng cha.

Khi cả hai điều kiện sau đều đúng, cơ sở dữ liệu sẽ khóa toàn bộ bảng trên bảng con:

- Không có chỉ mục nào tồn tại trên cột khóa ngoại của bảng con.
- Một phiên sửa đổi khóa chính trong bảng cha (ví dụ: xóa một hàng hoặc sửa đổi các thuộc tính khóa chính) hoặc hợp nhất các hàng vào bảng cha.

**Ghi chú:**

Việc chèn vào bảng cha không thu được các khóa bảng chặn ngăn DML trên bảng con. Trong trường hợp chèn, cơ sở dữ liệu sẽ lấy một khóa trên bảng con để ngăn các thay đổi về cấu trúc nhưng không ngăn được các sửa đổi của các hàng hiện có hoặc mới được thêm vào.

Giả sử bảng hr.departments là bảng mẹ của hr.employees, chứa khóa ngoại không được lập chỉ mục mscstaff.department\_id. Hình dưới đây cho thấy một phiên sửa đổi các thuộc tính khóa chính của bộ phận 60 trong bảng bộ phận.

Hình 9-3 Cơ chế khóa bằng khóa ngoại không được lập chỉ mục

**Khóa gốc**

Khóa chính của  
bảng được tham chiếu

Bảng được tham chiếu hoặc cha mẹ



Bảng CÁC BAN

DEPARTMENT_ID	DEPARTMENT_NAME	QUẢN LÝ_ID	LOCATION_ID
60	NÓ	103	1400
90	Điều hành	100	1700

Chứa được lập chỉ mục  
Khóa ngoại (các  
giá trị trong bảng phụ  
thuộc phải khớp với giá  
trị trong khóa duy nhất  
hoặc khóa chính của  
bảng được tham chiếu)

Bảng phụ thuộc hoặc bảng con



Bảng NHÂN VIÊN

MÃ HIỆU CÔNG NHÂN	LAST_NAME	EMAIL	HIRE_DATE	CÔNG VIỆC_ID	QUẢN LÝ_ID BỘ	PHẦN_ID
100	Nhà vua	TRUOT TRUOT	17-06-87	AD_PRES		90
101	Kochhar	NKOCHAR	21-09-89	AD_VP	100	90
102	De Hann	LDEHANN	13-01-93	AD_VP	100	90
103	Hunold	AHUNOLD	03-01-90	IT_PROG	102	60



Đã có được khóa toàn bộ bảng



Đã có được khóa hàng độc quyền (TX)



Khóa chính đã được sửa đổi

Khối lá chỉ mục

10, Ồn  
ào 20, Ồn ào  
...  
60, Ồn ào

Trong Hình 9-3, cơ sở dữ liệu thu được khóa bảng đầy đủ đối với nhân viên trong quá trình sửa đổi khóa chính của bộ phận 60. Khóa này cho phép các phiên khác truy vấn nhưng không cập nhật bảng nhân viên. Ví dụ: phiên không thể cập nhật số điện thoại của nhân viên. Khóa bảng đối với nhân viên sẽ được giải phóng ngay sau khi quá trình sửa đổi khóa chính trên bảng phòng ban hoàn tất. Nếu có nhiều hàng trong các phòng ban

trải qua các sửa đổi khóa chính, sau đó khóa bảng đối với nhân viên sẽ được lấy và phát hành một lần cho mỗi hàng được sửa đổi trong các phòng ban.



Ghi chú:

DML trên bảng con không yêu cầu khóa bảng trên bảng cha.

## Khóa và khóa ngoại được lập chỉ mục

Cơ sở dữ liệu không có được khóa bảng đầy đủ trên bảng con khi cột khóa ngoại trong bảng con được lập chỉ mục và phiên sửa đổi khóa chính trong bảng cha (ví dụ: xóa một hàng hoặc sửa đổi các thuộc tính khóa chính) hoặc hợp nhất các hàng vào bảng cha.

Khóa trên bảng cha sẽ ngăn các giao dịch có được các khóa bảng độc quyền, nhưng không ngăn DML trên bảng cha hoặc bảng con trong quá trình sửa đổi khóa chính. Tình huống này thích hợp hơn nếu các sửa đổi khóa chính xảy ra trên bảng cha trong khi các cập nhật xảy ra trên bảng con.


**Hình 9-4** hiển thị các nhân viên bảng con với cột Department\_id được lập chỉ mục. Một giao dịch xóa bộ phận 280 khỏi các bộ phận. Việc xóa này không khiến cơ sở dữ liệu có được khóa bảng đầy đủ trên bảng nhân viên như trong kịch bản được mô tả trong "Khóa và khóa ngoại không được lập chỉ mục".


Hình 9-4 Cơ chế khóa bằng khóa ngoại được lập chỉ mục

Khóa gốc

Khóa chính của  
bảng được tham chiếu

Bảng được tham chiếu hoặc cha mẹ

 Bảng CÁC BAN

DEPARTMENT_ID	DEPARTMENT_NAME	QUẢN LÝ_ID	LOCATION_ID
60	NÓ	103	1400
90	Điều hành	100	1700
 280	Lập kế hoạch sự kiện		1700




Đã lập chỉ mục

Khóa ngoại (các  
giá trị trong bảng phụ  
thuộc phải khớp với giá  
trị trong khóa duy nhất  
hoặc khóa chính của  
bảng được tham chiếu)


Bảng phụ thuộc hoặc bảng con

**Bảng NHÂN VIÊN**

MÃ HIỆU CÔNG NHÂN	LAST_NAME	EMAIL	HIRE_DATE	CÔNG VIỆC_ID	QUẢN LÝ_ID BỘ	NHÂN_ID
100	Nhà vua	TRUOT TRUOT	17-06-87	AD_PRES		90
101	Kochhar	NKOCHAR	21-09-89	AD_VP	100	90
102	De Hann	LDEHANN	13-01-93	AD_VP	100	90
103	Hunold	AHUNOLD	03-01-90	IT_PROG	102	60

	Đã có được khóa toàn bộ bảng
	Đã có được khóa hàng độc quyền (TX)
	Hàng đang bị xóa

Khởi lá chỉ mục

10, Ồn ào 20, Ồn ào ...
 280, Ồn ào

Nếu bảng con chỉ định **BẬT XÓA CASCADE** thì việc xóa khỏi bảng cha có thể dẫn đến việc xóa khỏi bảng con. Ví dụ: việc xóa phòng ban 280 có thể dẫn đến việc xóa hồ sơ của nhân viên đối với nhân viên trong phòng ban bị xóa. Trong trường hợp này, các quy tắc chờ và khóa cũng giống như khi bạn xóa các hàng khỏi bảng con sau khi xóa các hàng khỏi bảng cha.



Xem thêm:

- "Ràng buộc khóa ngoại"
- "Giới thiệu về chỉ mục"

## Khóa DDL

Khóa từ điển dữ liệu (DDL) bảo vệ định nghĩa của đối tượng lược đồ trong khi thao tác DDL đang diễn ra tác động lên hoặc tham chiếu đến đối tượng.

Chỉ các đối tượng lược đồ riêng lẻ được sửa đổi hoặc tham chiếu mới bị khóa trong các hoạt động DDL. Cơ sở dữ liệu không bao giờ khóa toàn bộ từ điển dữ liệu.

Cơ sở dữ liệu Oracle tự động mua khóa DDL thay mặt cho bất kỳ giao dịch DDL nào yêu cầu khóa đó. Người dùng không thể yêu cầu khóa DDL một cách rõ ràng. Ví dụ: nếu người dùng tạo một **thủ tục lưu trữ** thì Cơ sở dữ liệu Oracle sẽ tự động thu được các khóa DDL cho tất cả các đối tượng lược đồ được tham chiếu trong định nghĩa thủ tục. Khóa DDL ngăn không cho các đối tượng này bị thay đổi hoặc bị loại bỏ trước khi quá trình biên dịch thủ tục hoàn tất.

### Khóa DDL độc quyền

Khóa DDL độc quyền ngăn các phiên khác có được khóa DDL hoặc DML.

Hầu hết các hoạt động DDL yêu cầu khóa DDL độc quyền cho một tài nguyên để ngăn chặn sự can thiệp mang tính hủy diệt với các hoạt động DDL khác có thể sửa đổi hoặc tham chiếu cùng một đối tượng lược đồ. Ví dụ: DROP TABLE không được phép xóa bảng trong khi ALTER TABLE đang thêm một cột vào bảng đó và ngược lại.

Khóa DDL độc quyền tồn tại trong suốt thời gian thực thi câu lệnh DDL và cam kết tự động. Trong quá trình thu nhận một khóa DDL độc quyền, nếu một khóa DDL khác được giữ trên đối tượng lược đồ bằng một thao tác khác thì việc thu nhận sẽ đợi cho đến khi khóa DDL cũ hơn được giải phóng rồi mới tiếp tục.



#### Xem thêm:

"**Chia sẻ khóa DDL**" mô tả các tình huống không yêu cầu khóa độc quyền để ngăn chặn nhiều mang tính hủy diệt

### Chia sẻ khóa DDL

Khóa DDL chia sẻ cho một tài nguyên sẽ ngăn chặn sự can thiệp có tính hủy diệt với các hoạt động DDL xung đột, nhưng cho phép xử lý đồng thời dữ liệu cho các hoạt động DDL tương tự.

Ví dụ: khi câu lệnh TẠO THỦ TỤC được chạy, giao dịch chứa sẽ nhận được các khóa DDL chia sẻ cho tất cả các bảng được tham chiếu. Các giao dịch khác có thể đồng thời tạo các thủ tục tham chiếu đến cùng một bảng và thu được các khóa DDL chia sẻ đồng thời trên cùng một bảng, nhưng không giao dịch nào có thể có được một khóa DDL độc quyền trên bất kỳ bảng được tham chiếu nào.

Khóa DDL chia sẻ tồn tại trong suốt thời gian thực thi câu lệnh DDL và cam kết tự động. Do đó, một giao dịch có khóa DDL chia sẻ được đảm bảo rằng định nghĩa của đối tượng lược đồ được tham chiếu vẫn không đổi trong suốt giao dịch.

### Khóa phân tích có thể phá vỡ

Khóa phân tích cú pháp được giữ bởi một câu lệnh SQL hoặc đơn vị chương trình PL/SQL cho mỗi đối tượng lược đồ mà nó tham chiếu.



Khóa phân tích cú pháp được thu thập để **vùng SQL chia sẻ** liên quan có thể bị vô hiệu hóa nếu một đối tượng được tham chiếu bị thay đổi hoặc loại bỏ. Khóa phân tích cú pháp được gọi là khóa phân tích cú pháp có thể phá vỡ vì nó không cho phép bất kỳ hoạt động DDL nào và có thể bị phá vỡ để cho phép các hoạt động DDL xung đột.

Khóa phân tích cú pháp được lấy trong **nhóm chia sẻ** trong giai đoạn phân tích cú pháp thực thi câu lệnh SQL. Khóa được giữ miễn là vùng SQL dùng chung cho câu lệnh đó vẫn còn trong nhóm dùng chung.



Xem thêm:

"Bể bơi chung"

## Khóa hệ thống

Cơ sở dữ liệu Oracle sử dụng nhiều loại khóa hệ thống khác nhau để bảo vệ cấu trúc bộ nhớ và cơ sở dữ liệu nội bộ. Người dùng không thể truy cập được các cơ chế này vì người dùng không có quyền kiểm soát sự xuất hiện hoặc thời lượng của chúng.

## chốt

Chốt là một cơ chế tuần tự hóa cấp thấp, đơn giản, điều phối quyền truy cập của nhiều người dùng vào các cấu trúc dữ liệu, đối tượng và tệp được chia sẻ.

Chốt bảo vệ tài nguyên bộ nhớ dùng chung khỏi bị hỏng khi được nhiều tiến trình truy cập. Cụ thể, chốt bảo vệ cấu trúc dữ liệu khỏi các tình huống sau:

- Sửa đổi đồng thời bởi nhiều phiên
- Được đọc bởi một phiên

trong khi được sửa đổi bởi một phiên khác

- Phân bổ (hết thời gian) bộ nhớ trong khi được

truy cập Thông thường, một chốt duy nhất bảo vệ nhiều đối tượng trong SGA. Ví

dụ: các quy trình nền như DBW và LGWR phân bổ bộ nhớ từ **nhóm dùng chung** để tạo cấu trúc dữ liệu. Để phân bổ bộ nhớ này, các quy trình này sử dụng chốt nhóm chung để tuần tự hóa quyền truy cập để ngăn hai quy trình cố gắng kiểm tra hoặc sửa đổi nhóm chung cùng một lúc. Sau khi bộ nhớ được phân bổ, các quy trình khác có thể cần truy cập vào các khu vực nhóm dùng chung như bộ **đệm thư viện**, cần thiết để phân tích cú pháp. Trong trường hợp này, các tiến trình chỉ chốt bộ nhớ đệm của thư viện chứ không phải toàn bộ nhóm dùng chung.

Không giống như các chốt enqueue như khóa hàng, chốt không cho phép các phiên xếp hàng.

Khi một chốt có sẵn, phiên đầu tiên yêu cầu chốt sẽ có được quyền truy cập độc quyền vào nó. Hiện tượng **chốt quay** xảy ra khi một tiến trình liên tục yêu cầu chốt trong một vòng lặp, trong khi **trạng thái ngủ chốt** xảy ra khi một tiến trình giải phóng CPU trước khi gia hạn yêu cầu chốt.

Thông thường, quy trình của Oracle thu được chốt trong thời gian cực ngắn khi thao tác hoặc xem xét cấu trúc dữ liệu. Ví dụ: trong khi xử lý cập nhật lương của một nhân viên, cơ sở dữ liệu có thể lấy và giải phóng hàng nghìn chốt. Việc triển khai các chốt phụ thuộc vào hệ điều hành, đặc biệt là về việc liệu một quá trình có chờ một chốt hay không và trong bao lâu.

Việc tăng chốt có nghĩa là giảm tính đồng thời. Ví dụ: các thao tác **phân tích cú pháp cứng** quá mức sẽ tạo ra sự tranh chấp về chốt bộ nhớ đệm của thư viện. Chế độ xem V\$LATCH

chứa số liệu thống kê sử dụng chốt chi tiết cho từng chốt, bao gồm số lần mỗi chốt được yêu cầu và chờ đợi.



Xem thêm:

- "Phân tích cú pháp SQL"
- Tham khảo cơ sở dữ liệu Oracle để tìm hiểu về V\$LATCH
- Hướng dẫn điều chỉnh hiệu suất cơ sở dữ liệu Oracle để tìm hiểu về sự kiện chờ số liệu thống kê

## Mutexes

Đối tượng loại trừ lẫn nhau (mutex) là một cơ chế cấp thấp giúp ngăn chặn một đối tượng trong bộ nhớ bị cũ hoặc bị hỏng khi được truy cập bởi các tiến trình đồng thời. Một mutex tương tự như một chốt, nhưng trong khi một chốt thường bảo vệ một nhóm đối tượng thì một mutex chỉ bảo vệ một đối tượng.

Mutexes cung cấp một số lợi ích:

- Một mutex có thể làm giảm khả năng tranh chấp.  
Vì chốt bảo vệ nhiều đối tượng nên nó có thể trở thành nút cổ chai khi các tiến trình cố gắng truy cập đồng thời vào bất kỳ đối tượng nào trong số này. Bằng cách tuần tự hóa quyền truy cập vào một đối tượng riêng lẻ thay vì một nhóm, mutex sẽ tăng tính khả dụng.
- Một mutex tiêu thụ ít bộ nhớ hơn một chốt.
- Khi ở chế độ chia sẻ, một mutex cho phép tham chiếu đồng thời trong nhiều phiên.

## Khóa nội bộ

Khóa bên trong là các cơ chế cấp cao hơn, phức tạp hơn so với chốt và mutex và phục vụ nhiều mục đích khác nhau.

Cơ sở dữ liệu sử dụng các loại khóa nội bộ sau:

- Khóa bộ đệm từ điển

Các khóa này có thời hạn rất ngắn và được giữ trên các mục trong bộ đệm từ điển trong khi các mục này đang được sửa đổi hoặc sử dụng. Chúng đảm bảo rằng các câu lệnh được phân tích cú pháp sẽ không thấy các định nghĩa đối tượng không nhất quán. Khóa bộ đệm từ điển có thể được chia sẻ hoặc độc quyền. Các khóa chia sẻ được giải phóng khi quá trình phân tích cú pháp hoàn tất, trong khi các khóa độc quyền được giải phóng khi hoạt động DDL hoàn tất.

- Khóa quản lý tập tin và nhật ký

Những khóa này bảo vệ các tập tin khác nhau. Ví dụ: khóa bên trong bảo vệ **tệp điều khiển** để chỉ một tiến trình tại một thời điểm có thể thay đổi tệp đó. Một khóa khác điều phối việc sử dụng và lưu trữ các tệp nhật ký làm lại trực tuyến. Các tệp dữ liệu bị khóa để đảm bảo rằng nhiều phiên bản gắn kết cơ sở dữ liệu ở chế độ chia sẻ hoặc một phiên bản gắn kết cơ sở dữ liệu đó ở chế độ độc quyền. Vì khóa tệp và nhật ký cho biết trạng thái của tệp nên các khóa này nhất thiết phải được giữ trong thời gian dài.

- Không gian bảng và hoàn tác khóa phân đoạn

Những khóa này bảo vệ không gian bảng và hoàn tác các phân đoạn. Ví dụ: tất cả các phiên bản truy cập cơ sở dữ liệu phải đồng ý về việc một vùng bảng là trực tuyến hay ngoại tuyến. Các phân đoạn hoàn tác bị khóa để chỉ một phiên bản cơ sở dữ liệu có thể ghi vào một phân đoạn.



Xem thêm:

"Bộ đệm từ điển dữ liệu"

## Tổng quan về khóa dữ liệu thủ công

Bạn có thể ghi đè thủ công các cơ chế khóa mặc định của Cơ sở dữ liệu Oracle.

Cơ sở dữ liệu Oracle thực hiện khóa tự động để đảm bảo tính đồng thời của dữ liệu, tính toàn vẹn của dữ liệu và tính nhất quán khi đọc ở cấp độ câu lệnh. Tuy nhiên, việc ghi đè khóa mặc định rất hữu ích trong các trường hợp như sau:

- Các ứng dụng yêu cầu tính nhất quán khi đọc ở cấp độ giao dịch hoặc các lần đọc lặp lại.

Trong trường hợp này, các truy vấn phải tạo ra dữ liệu nhất quán trong suốt thời gian giao dịch, không phản ánh những thay đổi của các giao dịch khác. Bạn có thể đạt được tính nhất quán khi đọc ở cấp độ giao dịch bằng cách sử dụng khóa rõ ràng, giao dịch chỉ đọc, giao dịch có thể tuần tự hóa hoặc bằng cách ghi đè khóa mặc định.

- Ứng dụng yêu cầu giao dịch có quyền truy cập đọc quyền vào tài nguyên để giao dịch không phải đợi các giao dịch khác hoàn tất.

Bạn có thể ghi đè khóa tự động của Cơ sở dữ liệu Oracle ở cấp độ phiên hoặc giao dịch. Ở cấp độ phiên, một phiên có thể đặt mức cách ly giao dịch được yêu cầu bằng câu lệnh `ALTER SESSION`. Ở cấp độ giao dịch, các giao dịch bao gồm các câu lệnh SQL sau sẽ ghi đè khóa mặc định của Cơ sở dữ liệu Oracle:

- Câu lệnh `THIẾT LẬP MỨC ĐỘ CÁCH LY GIAO DỊCH`
- Câu lệnh `LOCK TABLE` (khóa một bảng hoặc khi được sử dụng với các dạng xem, bảng cơ sở)
- Câu lệnh `SELECT ... FOR UPDATE`

Các khóa có được từ các câu lệnh trước đó sẽ được giải phóng sau khi giao dịch kết thúc hoặc quay trở lại điểm lưu trữ sẽ giải phóng chúng.

Nếu khóa mặc định của Cơ sở dữ liệu Oracle bị ghi đè ở bất kỳ cấp độ nào thì quản trị viên cơ sở dữ liệu hoặc nhà phát triển ứng dụng phải đảm bảo rằng quy trình khóa ghi đè hoạt động chính xác. Các thủ tục khóa phải đáp ứng các tiêu chí sau: tính toàn vẹn của dữ liệu được đảm bảo, tính đồng thời của dữ liệu có thể chấp nhận được và không thể xảy ra tình trạng bế tắc hoặc được xử lý thích hợp.

**Xem thêm:**

- Tài liệu tham khảo ngôn ngữ SQL của cơ sở dữ liệu Oracle để biết các mô tả về LOCK BẢNG và CHỌN
- Hướng dẫn phát triển cơ sở dữ liệu Oracle để tìm hiểu cách khóa thủ công  
những cái bàn

## Tổng quan về khóa do người dùng xác định

Với dịch vụ Quản lý khóa cơ sở dữ liệu Oracle, bạn có thể xác định khóa của riêng mình cho một ứng dụng cụ thể.

Ví dụ: bạn có thể tạo một khóa để tuần tự hóa quyền truy cập vào nhật ký tin nhắn trên hệ thống tệp. Vì khóa người dùng dành riêng giống như khóa Cơ sở dữ liệu Oracle nên nó có tất cả chức năng khóa Cơ sở dữ liệu Oracle bao gồm cả phát hiện khóa chết. Khóa người dùng không bao giờ xung đột với khóa Cơ sở dữ liệu Oracle, vì chúng được xác định bằng tiền tố UL.

Các dịch vụ Quản lý khóa cơ sở dữ liệu Oracle có sẵn thông qua các thủ tục trong gói DBMS\_LOCK . Bạn có thể bao gồm các câu lệnh trong khối PL/SQL:

- Yêu cầu một loại khóa cụ thể
- Đặt cho khóa

một tên duy nhất có thể nhận biết được trong một quy trình khác trong cùng một trường hợp hoặc trong một trường hợp khác

- Thay đổi kiểu khóa
- Mở khóa

**Xem thêm:**

- Hướng dẫn phát triển cơ sở dữ liệu Oracle để biết thêm thông tin về Oracle Dịch vụ quản lý khóa cơ sở dữ liệu
- Tham chiếu các loại và gói PL/SQL của cơ sở dữ liệu Oracle để biết thông tin về DBMS\_LOCK

# 10

## Giao dịch

Chương này định nghĩa một giao dịch và mô tả cách cơ sở dữ liệu xử lý các giao dịch.

Chương này có các phần sau:

- [Giới thiệu về Giao dịch](#)
- [Tổng quan về Kiểm soát giao dịch](#)
- [Tổng quan về Bảo vệ giao dịch](#)
- [Tổng quan về tính liên tục của ứng dụng](#)
- [Tổng quan về giao dịch tự động](#)
- [Tổng quan về giao dịch phân phối](#)

## Giới thiệu về giao dịch

Giao dịch là một đơn vị công việc logic, nguyên tử có chứa một hoặc nhiều câu lệnh SQL.

Một giao dịch nhóm các câu lệnh SQL sao cho tất cả chúng đều được cam kết, nghĩa là chúng được áp dụng cho cơ sở dữ liệu hoặc tất cả được khôi phục, nghĩa là chúng được hoàn tác khỏi cơ sở dữ liệu. Cơ sở dữ liệu Oracle gán cho mỗi giao dịch một mã định danh duy nhất được gọi là [ID giao dịch](#).

Tất cả các giao dịch của Oracle đều tuân theo các thuộc tính cơ bản của giao dịch cơ sở dữ liệu, được gọi là [thuộc tính ACID](#). ACID là viết tắt của các từ sau:

- Tính nguyên tử

Tất cả các nhiệm vụ của một giao dịch đều được thực hiện hoặc không có nhiệm vụ nào được thực hiện. Không có giao dịch một phần. Ví dụ: nếu một giao dịch bắt đầu cập nhật 100 hàng nhưng hệ thống bị lỗi sau 20 lần cập nhật thì cơ sở dữ liệu sẽ khôi phục các thay đổi đối với 20 hàng này.

- Tính nhất quán

Giao dịch đưa cơ sở dữ liệu từ trạng thái nhất quán này sang trạng thái nhất quán khác. Ví dụ: trong một giao dịch ngân hàng ghi nợ tài khoản tiết kiệm và ghi có vào tài khoản séc, lỗi không được khiến cơ sở dữ liệu chỉ ghi có một tài khoản, điều này sẽ dẫn đến dữ liệu không nhất quán.

- Sự cách ly

Hiệu lực của một giao dịch không được hiển thị đối với các giao dịch khác cho đến khi giao dịch được thực hiện. Ví dụ: một người dùng đang cập nhật bảng hr.employees không thấy những thay đổi không được cam kết đối với nhân viên do người dùng khác thực hiện đồng thời. Do đó, người dùng có cảm giác như các giao dịch đang được thực hiện nối tiếp.

- Độ bền

Những thay đổi được thực hiện bởi các giao dịch đã cam kết là vĩnh viễn. Sau khi giao dịch hoàn tất, cơ sở dữ liệu đảm bảo thông qua cơ chế khôi phục của nó rằng các thay đổi từ giao dịch không bị mất.

Việc sử dụng các giao dịch là một trong những điểm quan trọng nhất mà hệ thống quản lý cơ sở dữ liệu khác với hệ thống tệp.

## Giao dịch mẫu: Ghi nợ và ghi có tài khoản

Để minh họa khái niệm giao dịch, hãy xem xét cơ sở dữ liệu ngân hàng.

Khi khách hàng chuyển tiền từ tài khoản tiết kiệm sang tài khoản séc, giao dịch phải bao gồm ba thao tác riêng biệt:

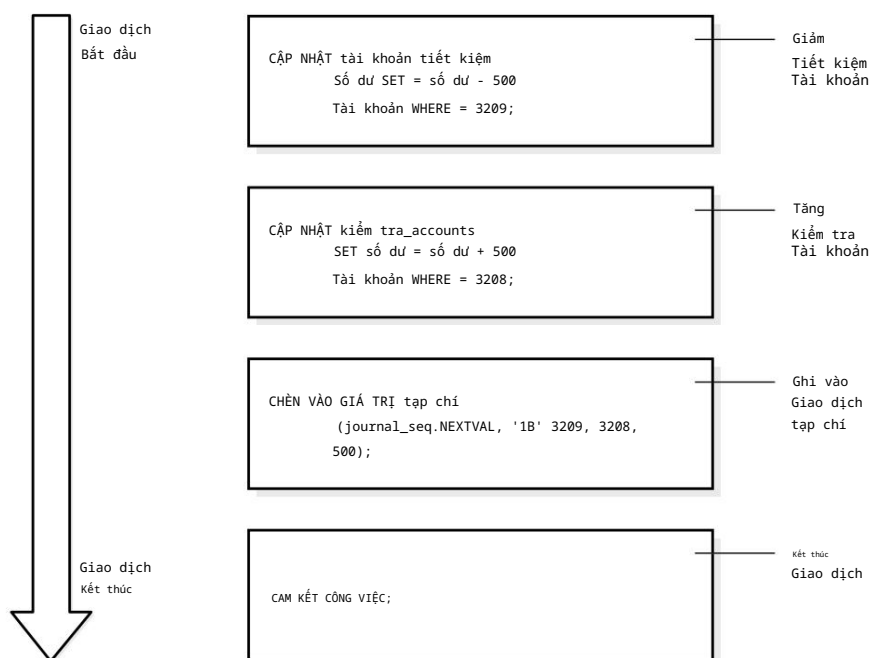
- Giảm tài khoản tiết kiệm • Tăng tài khoản séc • Ghi lại giao dịch vào nhật ký giao dịch

Cơ sở dữ liệu Oracle phải cho phép hai trường hợp. Nếu cả ba câu

lệnh SQL đều duy trì các tài khoản ở mức cân bằng hợp lý thì tác động của giao dịch có thể được áp dụng cho cơ sở dữ liệu. Tuy nhiên, nếu một vấn đề như không đủ tiền, số tài khoản không hợp lệ hoặc lỗi phần cứng khiến một hoặc hai câu lệnh trong giao dịch không thể hoàn thành thì cơ sở dữ liệu phải khôi phục toàn bộ giao dịch để số dư của tất cả các tài khoản là chính xác.

Hình ảnh sau đây minh họa một giao dịch ngân hàng. Câu lệnh đầu tiên trừ 500 USD từ tài khoản tiết kiệm 3209. Câu lệnh thứ hai cộng thêm 500 USD vào tài khoản séc 3208. Câu lệnh thứ ba chèn bản ghi chuyển khoản vào bảng nhật ký. Tuyên bố cuối cùng cam kết giao dịch.

Hình 10-1 Giao dịch ngân hàng



## Cấu trúc của một giao dịch

Một giao dịch cơ sở dữ liệu bao gồm một hoặc nhiều câu lệnh.

Cụ thể, một giao dịch bao gồm một trong những điều sau đây:

- Một hoặc nhiều câu lệnh ngôn ngữ thao tác dữ liệu (DML) cùng nhau tạo thành một sự thay đổi nguyên tử đối với cơ sở dữ liệu
- Một câu lệnh ngôn ngữ định nghĩa dữ liệu (DDL)

Một giao dịch có sự bắt đầu và kết thúc.



Xem thêm:

- ["Tổng quan về các câu lệnh SQL"](#)
- Tài liệu tham khảo ngôn ngữ SQL của cơ sở dữ liệu Oracle cho một tài khoản thuộc các loại của các câu lệnh SQL

## Bắt đầu một giao dịch

Một giao dịch bắt đầu khi gặp câu lệnh SQL thực thi đầu tiên.

Câu **lệnh SQL thực thi** là câu lệnh SQL tạo ra các lệnh gọi đến **cơ sở dữ liệu**

**ví dụ**, bao gồm các câu lệnh DML và DDL và câu lệnh SET TRANSACTION .

Khi một giao dịch bắt đầu, Cơ sở dữ liệu Oracle chỉ định giao dịch đó cho một phân đoạn **dữ liệu hoàn tác** có sẵn để ghi lại các mục hoàn tác cho giao dịch mới. ID giao dịch không được phân bổ cho đến khi phân đoạn hoàn tác và vị trí **bảng giao dịch** được phân bổ, điều này xảy ra trong câu lệnh DML đầu tiên. ID giao dịch là duy nhất cho một giao dịch và đại diện cho số phân đoạn hoàn tác, vị trí và số thứ tự.

Ví dụ sau đây thực thi câu lệnh UPDATE để bắt đầu giao dịch và truy vấn V\$TRANSACTION để biết chi tiết về giao dịch:

```
SQL> CẬP NHẬT nhân viên SET lương=lương;
```

Đã cập nhật 107 hàng.

```
SQL> CHON XID NHƯ "txn id", XIDUSN NHƯ "hoàn tác phân đoạn", XIDSLOT NHƯ "khe cấm",
      2 XIDSQN NHƯ "seq", TÌNH TRẠNG NHƯ "trạng thái txn"
      3 TỪ GIAO DỊCH V$;
```

id txn	hoàn tác phân đoạn	chỗ	trạng thái txn seq
0600060037000000	6	6	55 HOẠT ĐỘNG



Xem thêm:

- ["Hoàn tác phân đoạn"](#)

## Kết thúc giao dịch

Một giao dịch có thể kết thúc trong những trường hợp khác nhau.

Giao dịch kết thúc khi xảy ra bất kỳ hành động nào sau đây:

- Người dùng đưa ra câu lệnh COMMIT hoặc ROLLBACK mà không có mệnh đề SAVEPOINT .

Trong một **cam kết**, người dùng yêu cầu rõ ràng hoặc ngầm định rằng những thay đổi trong giao dịch phải được thực hiện vĩnh viễn. Những thay đổi do giao dịch thực hiện là vĩnh viễn và chỉ hiển thị cho những người dùng khác sau khi giao dịch được thực hiện.

- Người dùng chạy lệnh DDL chẳng hạn như CREATE, DROP, RENAME hoặc ALTER.

Cơ sở dữ liệu đưa ra một câu lệnh COMMIT ẩn trước và sau mỗi câu lệnh DDL. Nếu giao dịch hiện tại chứa các câu lệnh DML thì Cơ sở dữ liệu Oracle trước tiên sẽ thực hiện giao dịch đó, sau đó chạy và cam kết câu lệnh DDL dưới dạng một giao dịch mới, một câu lệnh.

- Người dùng thoát bình thường khỏi hầu hết các tiện ích và công cụ Cơ sở dữ liệu Oracle, khiến giao dịch hiện tại được ngầm thực hiện. Hành vi cam kết khi người dùng ngắt kết nối phụ thuộc vào ứng dụng và có thể định cấu hình.



### Ghi chú:

Các ứng dụng phải luôn cam kết hoặc hoàn tác các giao dịch một cách rõ ràng trước khi chấm dứt chương trình.

- Một tiến trình client kết thúc một cách bất thường, khiến cho giao dịch bị ngầm định được khôi phục bằng cách sử dụng siêu dữ liệu được lưu trữ trong bảng giao dịch và phân đoạn hoàn tác.

Sau khi một giao dịch kết thúc, câu lệnh SQL thực thi tiếp theo sẽ tự động bắt đầu giao dịch sau. Ví dụ sau thực thi CẬP NHẬT để bắt đầu giao dịch, kết thúc giao dịch bằng câu lệnh ROLLBACK và sau đó thực thi CẬP NHẬT để bắt đầu giao dịch mới (lưu ý rằng ID giao dịch là khác nhau):

```
SQL> CẬP NHẬT nhân viên SET Lương=Lương; Đã cập nhật 107 hàng.
```

```
SQL> CHỌN XID, TÌNH TRẠNG TỪ GIAO DỊCH V$;
```

```
XID                                TRẠNG THÁI
```

```
0800090033000000 HOẠT ĐỘNG
```

```
SQL> HOÀN LAI;
```

Hoàn tất khôi phục.

```
SQL> CHỌN XID TỪ GIAO DỊCH V$;
```

không có hàng nào được chọn

```
SQL> CẬP NHẬT hr.employees SET Last_name=last_name;
```

Đã cập nhật 107 hàng.

```
SQL> CHỌN XID, TÌNH TRẠNG TỪ GIAO DỊCH V$;
```



XID

TRẠNG THÁI

0900050033000000 HOẠT ĐỘNG



Xem thêm:

- "Giao dịch mẫu: Ghi nợ và ghi có tài khoản" để làm ví dụ về giao dịch kết thúc bằng một cam kết.
- Tham khảo ngôn ngữ SQL của cơ sở dữ liệu Oracle để tìm hiểu về COMMIT

## Tính nguyên tử ở cấp độ câu lệnh

Cơ sở dữ liệu Oracle hỗ trợ tính nguyên tử ở cấp độ câu lệnh, có nghĩa là câu lệnh SQL là một đơn vị công việc cơ bản và hoàn toàn thành công hoặc thất bại hoàn toàn.

Một tuyên bố thành công khác với một giao dịch đã cam kết. Một câu lệnh SQL sẽ thực thi thành công nếu cơ sở dữ liệu phân tích cú pháp và chạy nó mà không gặp lỗi dưới dạng một đơn vị nguyên tử, giống như khi tất cả các hàng được thay đổi trong một bản cập nhật nhiều hàng.

Nếu một câu lệnh SQL gây ra lỗi trong quá trình thực thi thì nó không thành công và do đó tất cả các hiệu ứng của câu lệnh sẽ bị khôi phục. Hoạt động này là một **khôi phục cấp độ câu lệnh**. Hoạt động này có các đặc điểm sau:

- Một câu lệnh SQL không thành công sẽ chỉ làm mất công việc mà lẽ ra nó phải thực hiện đã tự mình thực hiện.

Tuyên bố không thành công không gây mất bất kỳ công việc nào trước đó trong giao dịch hiện tại. Ví dụ: nếu việc thực thi câu lệnh CẬP NHẬT thứ hai trong "Giao dịch mẫu: Ghi nợ và ghi có tài khoản" gây ra lỗi và bị khôi phục, thì công việc được thực hiện bởi câu lệnh CẬP NHẬT đầu tiên sẽ không được khôi phục. Câu lệnh CẬP NHẬT đầu tiên có thể được cam kết hoặc khôi phục một cách rõ ràng bởi

người dùng.

- Hiệu ứng của việc khôi phục giống như câu lệnh chưa bao giờ được chạy.

Bất kỳ tác dụng phụ nào của câu lệnh nguyên tử, chẳng hạn như trình kích hoạt được gọi khi thực thi câu lệnh, đều được coi là một phần của câu lệnh nguyên tử. Tất cả công việc được tạo ra như một phần của tuyên bố nguyên tử đều thành công hoặc không có công việc nào thành công.

Một ví dụ về lỗi gây ra khôi phục cấp câu lệnh là cố gắng chen **khóa chính trùng lặp**. Các câu lệnh SQL đơn lẻ liên quan đến **tình trạng bế tắc**, đó là sự cạnh tranh cho cùng một dữ liệu, cũng có thể gây ra tình trạng khôi phục ở cấp độ câu lệnh. Tuy nhiên, các lỗi được phát hiện trong quá trình phân tích cú pháp câu lệnh SQL, chẳng hạn như lỗi cú pháp, vẫn chưa được chạy và do đó không gây ra hiện tượng khôi phục ở cấp độ câu lệnh.



Xem thêm:

- "Phân tích cú pháp SQL"
- "Khóa và bế tắc"
- "Tổng quan về Trình kích hoạt"

## Số thay đổi hệ thống (SCN)

Số thay đổi hệ thống (SCN) là dấu thời gian nội bộ, logic được Cơ sở dữ liệu Oracle sử dụng.

SCN sắp xếp các sự kiện xảy ra trong cơ sở dữ liệu, điều này cần thiết để đáp ứng các thuộc tính ACID của giao dịch. Cơ sở dữ liệu Oracle sử dụng SCN để đánh dấu SCN trước đó tất cả các thay đổi đều được biết là có trên đĩa để quá trình khôi phục tránh áp dụng các thao tác làm lại không cần thiết. Cơ sở dữ liệu cũng sử dụng SCN để đánh dấu điểm không tồn tại hoạt động làm lại đối với một tập hợp dữ liệu để quá trình khôi phục có thể dừng lại.

SCN xảy ra theo trình tự tăng dần đều. Cơ sở dữ liệu Oracle có thể sử dụng SCN giống như đồng hồ vì SCN được quan sát chỉ ra một thời điểm hợp lý và các quan sát lặp lại trả về giá trị bằng hoặc lớn hơn. Nếu một sự kiện có SCN thấp hơn sự kiện khác thì sự kiện đó đã xảy ra ở thời điểm sớm hơn trong cơ sở dữ liệu. Một số sự kiện có thể chia sẻ cùng một SCN, nghĩa là chúng xảy ra cùng lúc trong cơ sở dữ liệu.

Mọi giao dịch đều có SCN. Ví dụ: nếu một giao dịch cập nhật một hàng thì cơ sở dữ liệu sẽ ghi lại SCN nơi cập nhật này xảy ra. Các sửa đổi khác trong giao dịch này có cùng SCN. Khi một giao dịch được thực hiện, cơ sở dữ liệu sẽ ghi lại SCN cho lần thực hiện này.

Cơ sở dữ liệu Oracle tăng SCN trong [khu vực toàn cầu của hệ thống \(SGA\)](#). Khi một giao dịch sửa đổi dữ liệu, cơ sở dữ liệu sẽ ghi SCN mới vào phân đoạn dữ liệu hoàn tác được gán cho giao dịch. Sau đó, quy trình ghi nhật ký sẽ ghi ngay bản ghi cam kết của giao dịch vào [nhật ký làm lại trực tuyến](#). Bản ghi cam kết có SCN duy nhất của giao dịch. Cơ sở dữ liệu Oracle cũng sử dụng SCN như một phần của cơ chế [khôi phục phiên bản](#) và [phục hồi phương tiện](#).



Xem thêm:

- ["Tổng quan về khôi phục phiên bản"](#)
- ["Sao lưu và phục hồi"](#)

## Tổng quan về kiểm soát giao dịch

Kiểm soát giao dịch là quản lý các thay đổi được thực hiện bởi các câu lệnh DML và nhóm các câu lệnh DML thành các giao dịch.

Nói chung, các nhà thiết kế ứng dụng quan tâm đến việc kiểm soát giao dịch để công việc được thực hiện theo các đơn vị logic và dữ liệu được giữ nhất quán.

Kiểm soát giao dịch bao gồm việc sử dụng các câu lệnh sau, như được mô tả trong ["Tuyên bố kiểm soát giao dịch"](#):

- Câu lệnh COMMIT kết thúc giao dịch hiện tại và thực hiện mọi thay đổi thực hiện trong giao dịch vĩnh viễn. COMMIT cũng xóa tất cả các điểm lưu trữ trong giao dịch và giải phóng các khóa giao dịch.
- Câu lệnh ROLLBACK đảo ngược công việc đã thực hiện trong giao dịch hiện tại; Nó làm cho tất cả các thay đổi dữ liệu kể từ COMMIT hoặc ROLLBACK cuối cùng bị loại bỏ. Các

Câu lệnh ROLLBACK TO SAVEPOINT hoàn tác các thay đổi kể từ điểm lưu trữ cuối cùng nhưng không kết thúc toàn bộ giao dịch.

- Câu lệnh SAVEPOINT xác định một điểm trong giao dịch mà sau này bạn có thể cuộn lại.

Phần trong [Bảng 10-1](#) minh họa các khái niệm cơ bản về kiểm soát giao dịch.

Bảng 10-1 Kiểm soát giao dịch

Phiên T	Giải trình
t0 LÀM;	Tuyên bố này kết thúc mọi giao dịch hiện có trong phiên họp.
t1 ĐẶT TÊN GIAO DỊCH 'sal_update';	Câu lệnh này bắt đầu một giao dịch và đặt tên cho nó <b>sal_update</b> .
t2 CẬP NHẬT nhân viên Mức Lương BỘ = 7000 Ở ĐẦU Last_name = 'Banda';	Tuyên bố này cập nhật mức lương cho Banda lên 7000.
t3 LƯU ĐIỂM after_banda_sal;	Câu lệnh này tạo ra một điểm lưu trữ có tên <b>after_banda_sal</b> , cho phép thay đổi trong giao dịch này được quay trở lại thời điểm này.
t4 CẬP NHẬT nhân viên Mức Lương BỘ = 12000 Ở ĐẦU Last_name = 'Greene';	Tuyên bố này cập nhật mức lương cho Greene lên 12000.
t5 LƯU ĐIỂM after_greene_sal;	Câu lệnh này tạo ra một điểm lưu trữ có tên <b>after_greene_sal</b> , cho phép thay đổi trong giao dịch này được quay trở lại thời điểm này.
t6 QUAY LẠI ĐIỂM SAVEPOINT after_banda_sal;	Câu lệnh này khôi phục giao dịch về t3, hoàn tác cập nhật lương của Greene ở mức t4. Sal_update giao dịch chưa kết thúc.
t7 CẬP NHẬT nhân viên Mức lương BỘ = 11000 Ở ĐẦU Last_name = 'Greene';	Tuyên bố này cập nhật mức lương cho Greene lên 11000 trong giao dịch <b>sal_update</b> .
t8 HOÀN LẠI;	Tuyên bố này khôi phục tất cả các thay đổi trong giao dịch <b>sal_update</b> , kết thúc giao dịch.
t9 THIẾT LẬP TÊN GIAO DỊCH 'sal_update2';	Tuyên bố này bắt đầu một giao dịch mới trong phiên và đặt tên là <b>sal_update2</b> .
t10 CẬP NHẬT nhân viên Mức lương BỘ = 7050 Ở ĐẦU Last_name = 'Banda';	Tuyên bố này cập nhật mức lương cho Banda lên 7050.
t11 CẬP NHẬT nhân viên Mức lương BỘ = 10950 Ở ĐẦU Last_name = 'Greene';	Tuyên bố này cập nhật mức lương của Greene lên 10950.

Bảng 10-1 (Tiếp theo) Kiểm soát giao dịch

Phiên	T	Giải thích
t12	LÀM;	Câu lệnh này cam kết tất cả các thay đổi được thực hiện trong giao dịch sal_update2, kết thúc giao dịch. Cam kết đảm bảo rằng những thay đổi được lưu trong tệp nhật ký làm lại trực tuyến.

**Xem thêm:**

Tham khảo ngôn ngữ SQL cơ sở dữ liệu Oracle để tìm hiểu về các câu lệnh kiểm soát giao dịch

## Tên giao dịch

Tên giao dịch là một thẻ tùy chọn, do người dùng chỉ định, dùng để nhắc nhở về công việc mà giao dịch đang thực hiện. Bạn đặt tên cho một giao dịch bằng câu lệnh SET NAME , nếu được sử dụng thì GIAO DỊCH ... câu lệnh này phải là câu lệnh đầu tiên của giao dịch. giao dịch.

Trong [Bảng 10-1](#), giao dịch đầu tiên có tên là sal\_update và giao dịch thứ hai có tên là sal\_update2.

Tên giao dịch mang lại những lợi ích sau:

- Việc giám sát các giao dịch dài hạn và giải quyết các giao dịch phân tán đáng nghi ngờ sẽ dễ dàng hơn.
- Bạn có thể xem tên giao dịch cùng với ID giao dịch trong ứng dụng. Ví dụ: quản trị viên cơ sở dữ liệu có thể xem tên giao dịch trong Oracle Enterprise Manager (Trình quản lý doanh nghiệp) khi giám sát hoạt động của hệ thống.
- Cơ sở dữ liệu ghi tên giao dịch vào bản ghi làm lại kiểm tra giao dịch, do đó bạn có thể sử dụng LogMiner để tìm kiếm một giao dịch cụ thể trong nhật ký làm lại.
- Bạn có thể sử dụng tên giao dịch để tìm một giao dịch cụ thể trong từ điển dữ liệu các lượt xem như V\$TRANSACTION.

**Xem thêm:**

- "[Trình quản lý doanh nghiệp Oracle](#)"
- Tham khảo cơ sở dữ liệu Oracle để tìm hiểu về V\$TRANSACTION
- Tài liệu tham khảo ngôn ngữ SQL của cơ sở dữ liệu Oracle để tìm hiểu về SET GIAO DỊCH

## Giao dịch đang hoạt động

Giao dịch đang hoạt động là giao dịch đã bắt đầu nhưng chưa được cam kết hoặc chưa được khôi phục.

Trong [Bảng 10-1](#), câu lệnh đầu tiên sửa đổi dữ liệu trong giao dịch `sal_update` là cập nhật lương của Banda. Từ khi thực hiện thành công bản cập nhật này cho đến khi câu lệnh `ROLLBACK` kết thúc giao dịch, giao dịch `sal_update` sẽ hoạt động.

Những thay đổi dữ liệu do một giao dịch thực hiện chỉ là tạm thời cho đến khi giao dịch được thực hiện hoặc được khôi phục. Trước khi giao dịch kết thúc, trạng thái của dữ liệu như trong bảng sau.

Bảng 10-2 Trạng thái dữ liệu trước khi giao dịch kết thúc

Tình trạng	Sự miêu tả	Để tìm hiểu thêm
Cơ sở dữ liệu Oracle có tạo ra thông tin hoàn tác trong SGA.	Dữ liệu hoàn tác chứa giá trị dữ liệu cũ được thay đổi bởi các câu lệnh SQL của giao dịch.	<a href="#">"Đọc tính nhất quán trong Đọc cách ly đã cam kết Mức độ"</a>
Cơ sở dữ liệu Oracle đã tạo lại thao tác làm lại trong bộ đệm nhật ký làm lại trực tuyến của SGA.	Bản ghi nhật ký làm lại chứa thay đổi đối với khối dữ liệu và thay đổi đối với khối hoàn tác.	<a href="#">"Làm lại bộ đệm nhật ký "</a>
Những thay đổi đã được thực hiện đối với bộ đệm cơ sở dữ liệu của SGA.	Dữ liệu thay đổi đối với một giao dịch đã cam kết, được lưu trữ trong bộ đệm cơ sở dữ liệu của SGA, không nhất thiết phải được người viết cơ sở dữ liệu (DBW) ghi ngay vào tệp dữ liệu . Việc ghi đĩa có thể xảy ra trước hoặc sau cam kết.	<a href="#">"Cache đệm cơ sở dữ liệu"</a>
Các hàng bị ảnh hưởng bởi thay đổi dữ liệu sẽ bị khóa.	Những người dùng khác không thể thay đổi dữ liệu trong các hàng bị ảnh hưởng và họ cũng không thể thấy những thay đổi không được cam kết.	<a href="#">"Tóm tắt về khóa Hành vi"</a>

## Điểm lưu

Điểm lưu trữ là điểm đánh dấu trung gian do người dùng khai báo trong bối cảnh giao dịch.

Trong nội bộ, điểm đánh dấu điểm lưu trữ sẽ phân giải thành SCN. Điểm lưu chia một giao dịch dài thành các phần nhỏ hơn.


Nếu bạn sử dụng điểm lưu trữ trong một giao dịch dài thì sau này bạn có tùy chọn quay lại công việc được thực hiện trước thời điểm hiện tại trong giao dịch nhưng sau điểm lưu trữ được khai báo trong giao dịch. Vì vậy, nếu mắc lỗi, bạn không cần phải gửi lại mọi câu lệnh. [Bảng 10-1](#) tạo điểm lưu trữ `after_banda_sal` để cập nhật mức lương Greene có thể được khôi phục về điểm lưu trữ này.

## Quay lại Savepoint

Việc quay lại điểm lưu trữ trong một giao dịch không được cam kết có nghĩa là hoàn tác mọi thay đổi được thực hiện sau điểm lưu trữ đã chỉ định, nhưng nó không có nghĩa là quay lại giao dịch đó.

Khi một giao dịch được khôi phục về điểm lưu trữ, chẳng hạn như khi `ROLLBACK TO SAVEPOINT after_banda_sal` được chạy trong [Bảng 10-1](#), điều sau đây sẽ xảy ra:

1. Cơ sở dữ liệu Oracle chỉ khôi phục các câu lệnh chạy sau điểm lưu trữ.
- Trong **Bảng 10-1**, ROLLBACK TO SAVEPOINT khiến **CẬP NHẬT** cho Greene được cuộn lại quay lại, nhưng không phải là **CẬP NHẬT** cho Banda.
2. Cơ sở dữ liệu Oracle bảo tồn điểm lưu trữ được chỉ định trong ROLLBACK TO SAVEPOINT nhưng tất cả các điểm lưu tiếp theo sẽ bị mất.
- Trong **Bảng 10-1**, ROLLBACK TO SAVEPOINT khiến điểm lưu trữ `after_greene_sal` thành bị lạc.
3. Cơ sở dữ liệu Oracle giải phóng tất cả các khóa bảng và hàng có được sau thời gian được chỉ định điểm lưu trữ nhưng vẫn giữ lại tất cả các khóa dữ liệu có được trước điểm lưu trữ.
- Giao dịch vẫn hoạt động và có thể được tiếp tục.

 Xem thêm:

- Tham khảo ngôn ngữ SQL của cơ sở dữ liệu Oracle để tìm hiểu về ROLLBACK và câu lệnh SAVEPOINT
- Tham khảo ngôn ngữ PL/SQL của Cơ sở dữ liệu Oracle để tìm hiểu về xử lý và kiểm soát giao dịch

## Giao dịch xếp hàng

Tùy thuộc vào tình huống, các giao dịch đang chờ tài nguyên bị khóa trước đó có thể vẫn bị chặn sau khi quay lại điểm lưu trữ.

Khi một giao dịch bị chặn bởi một giao dịch khác, nó sẽ xếp vào hàng đợi bị chặn chính giao dịch đó, do đó toàn bộ giao dịch chặn phải được cam kết hoặc khôi phục trong thời gian giao dịch bị chặn vẫn tiếp tục.

Trong kịch bản hiển thị trong bảng sau, phiên 1 sẽ quay lại điểm lưu trữ được tạo trước khi nó thực thi câu lệnh DML. Tuy nhiên, phiên 2 vẫn bị chặn bởi vì nó đang chờ giao dịch phiên 1 hoàn tất.

Bảng 10-3 Ví dụ quay lại Savepoint

T Phiên	1	Buổi 2	Buổi 3	Giải trình
t0	CẬP NHẬT nhân viên Mức lương BỘ = 7000 Ở ĐẦU họ_= 'Banda';			Buổi 1 bắt đầu một giao dịch. Các phiên đặt một khóa đọc quyền trên Hàng Banda (TX) và một khóa bàn đọc quyền (SX) trên bàn.
t1	LƯU ĐIỂM after_banda_sal;			Phiên 1 tạo ra một điểm lưu trữ được đặt tên <b>after_banda_sal</b> .
t2	CẬP NHẬT nhân viên Mức lương BỘ = 12000 Ở ĐẦU họ_= 'Xanh';			Phiên 1 khóa Hàng xanh .

Bảng 10-3 (Tiếp theo) Ví dụ khôi phục về Savepoint

T Phiên 1	Buổi 2	Buổi 3	Giải thích Phần
t3	CẬP NHẬT nhân viên Mức lương BỘ = 14000 Ở ĐẦU họ_= 'Xanh';		2 cố gắng cập nhật hàng Greene nhưng không lấy được khóa vì phiên 1 có khóa trên hàng này.  Không có giao dịch nào có bắt đầu ở buổi 2.
t4	HOÀN LẠI ĐỂ LƯU ĐIỂM after_banda_sal;		Phần 1 khôi phục bản cập nhật về mức lương cho Greene, đồng thời giải phóng khóa hàng cho Greene.  Khóa bảng thu được ở thời điểm t0 không được giải phóng.  Tại thời điểm này, phiên 2 vẫn bị chặn bởi phiên 1 vì phiên 2 xếp hàng vào giao dịch phiên 1, chưa hoàn thành.
t5		CẬP NHẬT nhân viên Mức lương BỘ = 11000 Ở ĐẦU họ_= 'Xanh';	Hàng Greene hiện đã được mở khóa, vì vậy phiên 3 sẽ nhận được khóa để cập nhật hàng Greene . Tuyên bố này bắt đầu một giao dịch trong phiên 3.
t6	LÀM;		Phiên 1 cam kết, kết thúc giao dịch của nó. Phiên 2 bây giờ là được xếp hàng để cập nhật hàng Greene đăng sau giao dịch trong phiên 3.



Xem thêm:

"Thời lượng khóa" để tìm hiểu thêm về thời điểm Cơ sở dữ liệu Oracle phát hành khóa

## Khôi phục giao dịch

Việc khôi phục một giao dịch không được cam kết sẽ hoàn tác mọi thay đổi đối với dữ liệu đã được các câu lệnh SQL thực hiện trong giao dịch đó.

Sau khi một giao dịch được khôi phục, ảnh hưởng của công việc được thực hiện trong giao dịch đó không còn tồn tại. Khi khôi phục toàn bộ giao dịch mà không tham chiếu bất kỳ điểm lưu trữ nào, Cơ sở dữ liệu Oracle thực hiện các hành động sau:

- Hoàn tác tất cả các thay đổi được thực hiện bởi tất cả các câu lệnh SQL trong giao dịch bằng cách sử dụng các phân đoạn hoàn tác tương ứng

Mục nhập bảng giao dịch cho mọi giao dịch đang hoạt động chứa một con trỏ tới tất cả dữ liệu hoàn tác (theo thứ tự ứng dụng ngược lại) cho giao dịch. Cơ sở dữ liệu đọc dữ liệu từ phân đoạn hoàn tác, đảo ngược thao tác và sau đó đánh dấu mục hoàn tác là đã áp dụng. Vì vậy, nếu một giao dịch chèn một hàng thì việc khôi phục sẽ xóa nó. Nếu một giao dịch cập nhật một hàng thì việc khôi phục sẽ đảo ngược quá trình cập nhật. Nếu một giao dịch xóa một hàng thì việc khôi phục sẽ chèn lại nó. Trong [Bảng 10-1](#), ROLLBACK đảo ngược các cập nhật về lương của Greene và Banda.

- Giải phóng tất cả các khóa dữ liệu được giữ bởi giao dịch
- Xóa tất cả

các điểm lưu trữ trong giao dịch Trong [Bảng 10-1](#),

ROLLBACK xóa điểm lưu trữ after\_banda\_sal. Điểm lưu trữ after\_greene\_sal đã bị xóa bởi câu lệnh ROLLBACK TO SAVEPOINT .

- Kết thúc giao dịch

Trong [Bảng 10-1](#), ROLLBACK để cơ sở dữ liệu ở trạng thái tương tự như sau khi COMMIT ban đầu được thực thi.

Thời lượng khôi phục là một hàm của lượng dữ liệu được sửa đổi.



Xem thêm:

- "Hoàn tác phân đoạn" •

Tham khảo ngôn ngữ SQL của cơ sở dữ liệu Oracle để tìm hiểu về ROLLBACK yêu cầu

## Cam kết giao dịch

Một cam kết kết thúc giao dịch hiện tại và thực hiện vĩnh viễn tất cả các thay đổi được thực hiện trong giao dịch.

Trong [Bảng 10-1](#), giao dịch thứ hai bắt đầu bằng sal\_update2 và kết thúc bằng câu lệnh COMMIT rõ ràng . Những thay đổi do hai câu lệnh CẬP NHẬT hiện được thực hiện vĩnh viễn.

Khi một giao dịch được thực hiện, các hành động sau sẽ xảy ra:

- Cơ sở dữ liệu tạo ra SCN cho COMMIT.

Bảng giao dịch nội bộ dành cho các bản ghi [không gian bảng hoàn tác](#) liên quan mà giao dịch đã cam kết. SCN duy nhất tương ứng của giao dịch được chỉ định và ghi lại trong bảng giao dịch.

- Quá trình ghi nhật ký (LGWR) ghi các mục nhật ký làm lại còn lại trong bộ đệm nhật ký làm lại vào nhật ký làm lại trực tuyến và ghi SCN giao dịch vào nhật ký làm lại trực tuyến. Sự kiện nguyên tử này cấu thành sự cam kết của giao dịch.
- Cơ sở dữ liệu Oracle giải phóng các khóa được giữ trên các hàng và bảng.

Những người dùng đang xếp hàng chờ đợi các khóa được giữ bởi giao dịch không được cam kết sẽ được phép tiếp tục công việc của họ.

- Cơ sở dữ liệu Oracle xóa các điểm lưu trữ.



Trong [Bảng 10-1](#), không có điểm lưu trữ nào tồn tại trong giao dịch `sal_update` nên không có điểm lưu trữ nào bị xóa.

- Cơ sở dữ liệu Oracle thực hiện việc [dọn dẹp cam kết](#).

Nếu các khối được sửa đổi chứa dữ liệu từ giao dịch đã cam kết vẫn còn trong SGA và nếu không có phiên nào khác sửa đổi chúng thì cơ sở dữ liệu sẽ xóa thông tin giao dịch liên quan đến khóa (mục nhập ITL) khỏi các khối.

Lý tưởng nhất là COMMIT xóa các khối để SELECT tiếp theo không phải thực hiện nhiệm vụ này. Nếu không có mục ITL nào tồn tại cho một hàng cụ thể thì nó không bị khóa. Nếu một mục nhập ITL tồn tại cho một hàng cụ thể thì nó có thể bị khóa, do đó phiên phải kiểm tra tiêu đề phân đoạn hoàn tác để xác định xem giao dịch quan tâm này có được thực hiện hay không. Nếu giao dịch quan tâm đã được cam kết thì phiên sẽ xóa khối và tạo ra thao tác làm lại. Tuy nhiên, nếu COMMIT đã xóa ITL trước đó thì việc kiểm tra và xóa là không cần thiết.



#### Ghi chú:

Vì việc dọn dẹp khối tạo ra thao tác làm lại nên một truy vấn có thể tạo ra thao tác làm lại và do đó khiến các khối được ghi trong điểm [kiểm tra tiếp theo](#).

- Cơ sở dữ liệu Oracle đánh dấu giao dịch đã hoàn tất.

Sau khi giao dịch được cam kết, người dùng có thể xem các thay đổi.

Thông thường, một cam kết là một hoạt động nhanh chóng, bất kể quy mô giao dịch. Tốc độ của một cam kết không tăng theo kích thước của dữ liệu được sửa đổi trong giao dịch. Phần dài nhất của cam kết là I/O đĩa vật lý được thực hiện bởi LGWR. Tuy nhiên, lượng thời gian mà LGWR sử dụng sẽ giảm đi do nó ghi dần dần nội dung của bộ đệm nhật ký làm lại ở chế độ nền.

Hành vi mặc định là để LGWR ghi lại thao tác làm lại vào nhật ký làm lại trực tuyến một cách đồng bộ và đối với các giao dịch là đợi thao tác làm lại được lưu vào bộ đệm trên đĩa trước khi trả lại cam kết cho người dùng. Tuy nhiên, để độ trễ cam kết giao dịch thấp hơn, nhà phát triển ứng dụng có thể chỉ định rằng việc làm lại được ghi không đồng bộ để các giao dịch không cần phải đợi việc làm lại trên đĩa và có thể quay lại từ lệnh gọi COMMIT ngay lập tức.



#### Xem thêm:

- "[Mức cô lập có thể tuân tự hóa](#)"
- "[Cơ chế khóa](#)"
- "[Tổng quan về các quy trình nền](#)" để biết thêm thông tin về LGWR
- Tham khảo các loại và gói PL/SQL của cơ sở dữ liệu Oracle để biết thêm thông tin về cam kết không đồng bộ

## Tổng quan về Bảo vệ giao dịch

Bảo vệ giao dịch là một API mà các ứng dụng có thể sử dụng để cung cấp tính bình thường của giao dịch, đó là khả năng cơ sở dữ liệu duy trì một cam kết được đảm bảo

kết quả cho biết liệu một giao dịch có được cam kết và hoàn thành hay không. Cơ sở dữ liệu Oracle cung cấp API cho JDBC mỏng, OCI, OCCI và ODP.Net.

Lỗi **có thể phục hồi** là do lỗi hệ thống bên ngoài, không phụ thuộc vào logic phiên ứng dụng đang thực thi. Các lỗi có thể phục hồi xảy ra sau sự cố ngừng hoạt động theo kế hoạch và ngoài kế hoạch của các quy trình, mạng, nút, bộ lưu trữ và cơ sở dữ liệu trên nền trước. Nếu sự cố ngừng hoạt động làm gián đoạn kết nối giữa ứng dụng khách và cơ sở dữ liệu thì ứng dụng đó sẽ nhận được thông báo lỗi ngắt kết nối. Giao dịch đang chạy khi kết nối bị hỏng được gọi là **giao dịch đang thực hiện**.

Để quyết định gửi lại giao dịch hay trả lại kết quả (đã cam kết hoặc không cam kết) cho khách hàng, ứng dụng phải xác định kết quả của giao dịch trên chuyến bay. Trước Cơ sở dữ liệu Oracle 12c, các thông báo cam kết được trả về cho máy khách không tồn tại lâu dài. Việc kiểm tra một giao dịch không đảm bảo rằng nó sẽ không được thực hiện sau khi được kiểm tra, tạo điều kiện cho các giao dịch trùng lặp và các hình thức sai lệch logic khác. Ví dụ: người dùng có thể làm mới trình duyệt web khi mua sách trực tuyến và bị tính phí hai lần cho cùng một cuốn sách.



#### Xem thêm:

- ["Giới thiệu về giao dịch"](#)
- [Hướng dẫn phát triển cơ sở dữ liệu Oracle để tìm hiểu về Transaction Guard](#)
- [Hướng dẫn triển khai và quản trị cụm ứng dụng thực của Oracle để tìm hiểu cách định cấu hình các dịch vụ cho Bộ bảo vệ giao dịch](#)

## Lợi ích của Bảo vệ giao dịch

Bắt đầu từ Cơ sở dữ liệu Oracle 12c, Transaction Guard cung cấp cho các ứng dụng một công cụ để xác định trạng thái của một giao dịch đang diễn ra sau khi ngừng hoạt động có thể phục hồi.

Sử dụng Transaction Guard, một ứng dụng có thể đảm bảo rằng giao dịch được thực hiện không quá một lần. Ví dụ: nếu ứng dụng hiệu sách trực tuyến xác định rằng cam kết đã gửi trước đó không thành công thì ứng dụng đó có thể gửi lại một cách an toàn.

Transaction Guard cung cấp một công cụ thực thi tối đa một lần để tránh ứng dụng thực hiện các lần gửi trùng lặp. Transaction Guard cung cấp kết quả đã biết cho mọi giao dịch.

Transaction Guard là một khả năng cốt lõi của Cơ sở dữ liệu Oracle. Tính liên tục của ứng dụng sử dụng Trình bảo vệ giao dịch khi che giấu sự cố ngừng hoạt động của người dùng cuối. Nếu không có Bộ bảo vệ giao dịch, việc thử lại ứng dụng sau khi xảy ra lỗi có thể khiến các giao dịch trùng lặp được thực hiện.

### Xem thêm:

- "Tổng quan về tính liên tục của ứng dụng" để tìm hiểu về ứng dụng tính liên tục, hoạt động với Transaction Guard để giúp các nhà phát triển đạt được tính khả dụng cao của ứng dụng
- Hướng dẫn phát triển cơ sở dữ liệu Oracle để tìm hiểu về Transaction Guard, bao gồm các loại giao dịch được hỗ trợ và đi kèm

## Cách thức hoạt động của Bảo vệ giao dịch

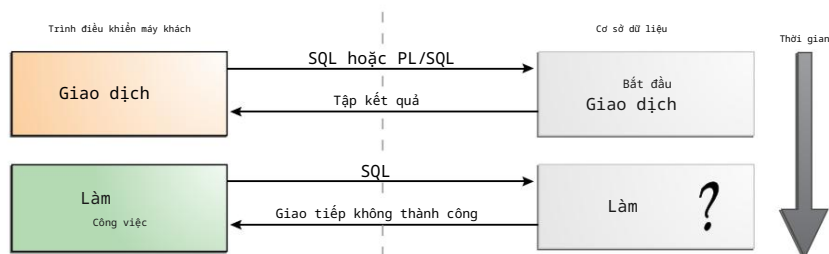
Phần này giải thích vấn đề mất thông báo cam kết và cách Transaction Guard sử dụng ID giao dịch logic để giải quyết vấn đề.

## Tin nhắn cam kết bị mất

Khi thiết kế cho idempotence, các nhà phát triển phải giải quyết vấn đề lỗi giao tiếp sau khi gửi các báo cáo cam kết. Thông báo cam kết không tồn tại trong cơ sở dữ liệu và do đó không thể truy xuất được sau khi thất bại.

Đồ họa sau đây thể hiện mức độ cao về sự tương tác giữa ứng dụng khách và cơ sở dữ liệu.

Hình 10-2 Thông báo cam kết bị mất



Trong trường hợp cam kết tiêu chuẩn, cơ sở dữ liệu cam kết một giao dịch và trả về thông báo thành công cho khách hàng. Trong [Hình 10-2](#), máy khách gửi một câu lệnh cam kết và nhận được một thông báo cho biết việc giao tiếp không thành công. Loại lỗi này có thể xảy ra vì một số lý do, bao gồm lỗi phiên bản cơ sở dữ liệu hoặc mất mạng. Trong trường hợp này, khách hàng không biết trạng thái của giao dịch.

Sau lỗi giao tiếp, cơ sở dữ liệu có thể vẫn đang chạy quá trình gửi và không biết rằng máy khách đã bị ngắt kết nối. Việc kiểm tra trạng thái giao dịch không đảm bảo rằng giao dịch đang hoạt động sẽ không được thực hiện sau khi được kiểm tra. Nếu khách hàng gửi lại cam kết vì thông tin lỗi thời này thì cơ sở dữ liệu có thể lặp lại giao dịch, dẫn đến lỗi logic.

## ID giao dịch hợp lý

Cơ sở dữ liệu Oracle giải quyết lỗi giao tiếp bằng cách sử dụng mã định danh duy nhất trên toàn cầu được gọi là ID giao dịch logic.

ID này chứa số phiên logic được phân bổ khi phiên kết nối lần đầu tiên và số cam kết đang chạy được cập nhật mỗi khi phiên cam kết hoặc quay lại.<sup>1</sup> Từ góc độ ứng dụng, ID giao dịch logic xác định duy nhất giao dịch cơ sở dữ liệu cuối cùng được gửi trên phiên đó không thành công.

Đối với mỗi chuyến đi khứ hồi từ máy khách trong đó một hoặc nhiều giao dịch được cam kết, cơ sở dữ liệu sẽ lưu trữ ID giao dịch logic. ID này có thể cung cấp tính bình thường của giao dịch cho các tương tác giữa ứng dụng và cơ sở dữ liệu cho mỗi chuyến đi khứ hồi cam kết dữ liệu.

Giao thức nhiều nhất một lần cho phép truy cập vào kết quả cam kết bằng cách yêu cầu cơ sở dữ liệu thực hiện những việc sau:

- Duy trì ID giao dịch hợp lý trong khoảng thời gian lưu giữ đã thỏa thuận để thử lại
- Duy trì ID giao dịch hợp lý trên cam kết

Trong khi giao dịch đang chạy, cả cơ sở dữ liệu và ứng dụng khách đều giữ ID giao dịch logic. Cơ sở dữ liệu cung cấp cho khách hàng một ID giao dịch logic khi xác thực, khi mượn từ nhóm kết nối và tại mỗi chuyến đi khứ hồi từ trình điều khiển máy khách thực hiện một hoặc nhiều thao tác cam kết.

Trước khi ứng dụng có thể xác định kết quả của giao dịch cuối cùng sau một lỗi có thể khôi phục, ứng dụng sẽ lấy ID giao dịch logic được giữ ở máy khách bằng cách sử dụng API Java, OCI, OCCI hoặc ODP.Net. Sau đó, ứng dụng sẽ gọi thủ tục PL/SQL DBMS\_APP\_CONT.GET\_LTXID\_OUTCOME với ID giao dịch logic để xác định kết quả của lần gửi cuối cùng: đã cam kết (đúng hoặc sai) và lệnh gọi của người dùng đã hoàn thành (đúng hoặc sai).

Khi sử dụng Transaction Guard, ứng dụng có thể phát lại các giao dịch khi lỗi có thể khắc phục được và giao dịch cuối cùng trong phiên chưa được cam kết. Ứng dụng có thể tiếp tục khi giao dịch cuối cùng được thực hiện và cuộc gọi của người dùng đã hoàn tất. Ứng dụng có thể sử dụng Transaction Guard để trả về kết quả đã biết cho khách hàng để khách hàng có thể quyết định hành động tiếp theo cần thực hiện.



#### Xem thêm:

- Hướng dẫn phát triển cơ sở dữ liệu Oracle để tìm hiểu về giao dịch logic ID
- Tham khảo các loại và gói PL/SQL của cơ sở dữ liệu Oracle để tìm hiểu thêm về thủ tục DBMS\_APP\_CONT.GET\_LTXID\_OUTCOME

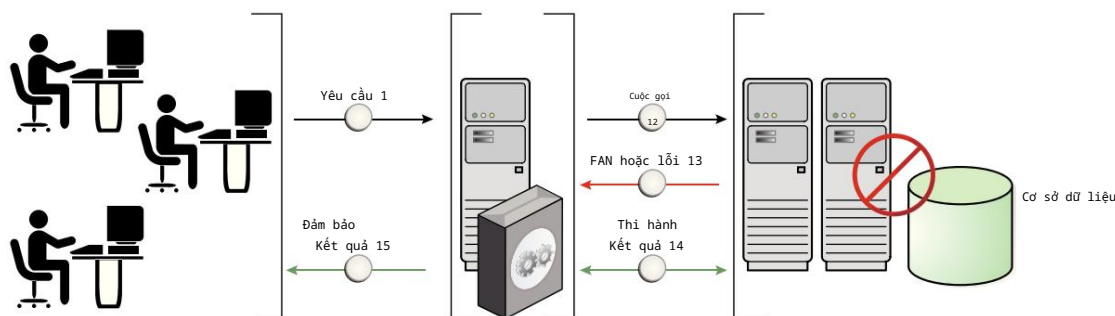
## Bảo vệ giao dịch: Ví dụ

Trong trường hợp này, thông báo cam kết bị mất do lỗi có thể phục hồi.

Bộ bảo vệ giao dịch sử dụng ID giao dịch logic để bảo toàn kết quả của câu lệnh COMMIT, đảm bảo rằng có một kết quả đã biết cho giao dịch.

<sup>1</sup> Đối với Cụm ứng dụng thực của Oracle (Oracle RAC), ID giao dịch logic bao gồm phiên bản cơ sở dữ liệu số làm tiền tố.

Hình 10-3 Kiểm tra trạng thái giao dịch logic



Trong **Hình 10-3**, cơ sở dữ liệu sẽ thông báo cho ứng dụng xem giao dịch đã được thực hiện hay chưa và liệu cuộc gọi cuối cùng của người dùng có hoàn thành hay không. Sau đó, ứng dụng có thể trả về kết quả cho người dùng cuối. Các khả năng là:

- Nếu giao dịch đã được thực hiện và lệnh gọi của người dùng hoàn thành thì ứng dụng có thể trả về kết quả cho người dùng cuối và tiếp tục.
- Nếu giao dịch đã được thực hiện nhưng lệnh gọi của người dùng không hoàn thành thì ứng dụng có thể trả về kết quả cho người dùng cuối kèm theo cảnh báo. Các ví dụ bao gồm liên kết bị mất hoặc số lượng hàng được xử lý bị mất. Một số ứng dụng phụ thuộc vào thông tin bổ sung, trong khi những ứng dụng khác thì không.
- Nếu lệnh gọi của người dùng không được thực hiện thì ứng dụng có thể trả lại thông tin này cho người dùng cuối hoặc gửi lại một cách an toàn. Giao thức được đảm bảo. Khi trạng thái cam kết trả về sai, lần gửi cuối cùng sẽ bị chặn thực hiện.

#### Xem thêm:

Hướng dẫn phát triển cơ sở dữ liệu Oracle để tìm hiểu cách sử dụng Transaction Guard

## Tổng quan về tính liên tục của ứng dụng

Tính liên tục của ứng dụng cố gắng che giấu sự cố ngừng hoạt động của các ứng dụng bằng cách phát lại các yêu cầu ứng dụng chưa hoàn chỉnh sau các lần ngừng hoạt động ngoài kế hoạch và theo kế hoạch. Trong ngữ cảnh này, yêu cầu là một đơn vị công việc từ ứng dụng.

Thông thường, một yêu cầu tương ứng với các câu lệnh DML và các lệnh gọi cơ sở dữ liệu khác của một yêu cầu web trên một kết nối cơ sở dữ liệu. Nói chung, một yêu cầu được phân định bằng các cuộc gọi được thực hiện giữa quá trình đăng ký và đăng ký kết nối cơ sở dữ liệu từ nhóm kết nối.

Phần này bao gồm các chủ đề sau:

- Lợi ích của tính liên tục của ứng dụng
- Kiến trúc liên tục của ứng dụng

## Lợi ích của tính liên tục của ứng dụng

Một vấn đề cơ bản đối với các nhà phát triển là làm thế nào để che giấu phiên cơ sở dữ liệu bị mất từ đầu đến cuối.  
người dùng.

Tính liên tục của ứng dụng cố gắng giải quyết vấn đề mất phiên bằng cách khôi phục phiên cơ sở dữ liệu khi bất kỳ thành phần nào làm gián đoạn cuộc trò chuyện giữa cơ sở dữ liệu và máy khách. Phiên cơ sở dữ liệu được khôi phục bao gồm tất cả các trạng thái, con trỏ, biến và giao dịch gần đây nhất khi có.

### Trường hợp sử dụng cho tính liên tục của ứng dụng

Trong trường hợp điển hình, khách hàng đã gửi yêu cầu tới cơ sở dữ liệu, cơ sở dữ liệu đã xây dựng cả trạng thái giao dịch và phi giao dịch.

Trạng thái tại máy khách vẫn hiện tại, có thể có dữ liệu đã nhập, dữ liệu trả về cũng như dữ liệu và biến được lưu trong bộ nhớ đệm. Tuy nhiên, trạng thái phiên cơ sở dữ liệu mà ứng dụng cần hoạt động bên trong sẽ bị mất.

Nếu yêu cầu của khách hàng đã bắt đầu một hoặc nhiều giao dịch thì ứng dụng sẽ gặp phải các khả năng sau:

- Nếu một cam kết đã được ban hành thì thông báo cam kết được trả về cho khách hàng sẽ không tồn tại lâu dài. Máy khách không biết liệu yêu cầu có được thực hiện hay không và nó đã đạt đến trạng thái xử lý phi giao dịch nào.
- Nếu một cam kết chưa được ban hành hoặc nếu nó được ban hành nhưng không được thực thi thì giao dịch đang diễn ra sẽ được khôi phục và phải được gửi lại bằng một phiên ở trạng thái chính xác.

Nếu việc phát lại thành công thì dịch vụ người dùng cơ sở dữ liệu dành cho các trường hợp ngừng hoạt động theo kế hoạch và ngoài kế hoạch sẽ không bị gián đoạn. Nếu cơ sở dữ liệu phát hiện những thay đổi trong dữ liệu được nhìn thấy và có khả năng được ứng dụng thực hiện thì việc phát lại sẽ bị từ chối. Không thử phát lại khi vượt quá thời gian cho phép để bắt đầu phát lại, ứng dụng sử dụng lệnh gọi bị hạn chế hoặc ứng dụng đã vô hiệu hóa rõ ràng tính năng phát lại bằng phương thức vô hiệu hóaReplay .



### Xem thêm:

Hướng dẫn triển khai và quản trị cụm ứng dụng thực của Oracle để tìm hiểu thêm về cách hoạt động liên tục của ứng dụng cho các phiên cơ sở dữ liệu

### Tính liên tục của ứng dụng để bảo trì theo kế hoạch

Tính liên tục của ứng dụng trong trường hợp ngừng hoạt động theo kế hoạch cho phép các ứng dụng tiếp tục hoạt động đối với các phiên cơ sở dữ liệu có thể được thoát hoặc di chuyển một cách đáng tin cậy.

Bảo trì theo lịch trình không cần làm gián đoạn công việc của ứng dụng. Tính liên tục của ứng dụng cung cấp thời gian làm việc tích cực để chuyển từ vị trí hiện tại sang vị trí mới hiện không bị ảnh hưởng bởi việc bảo trì. Khi kết thúc khoảng thời gian thoát, các phiên có thể vẫn còn trên phiên bản cơ sở dữ liệu nơi bảo trì được lên kế hoạch. Thay vì buộc phải ngắt kết nối các phiên này, tính liên tục của ứng dụng có thể chuyển các phiên này sang một trang web còn tồn tại và gửi lại mọi giao dịch đang diễn ra.

Với tính năng Liên tục ứng dụng được bật, cơ sở dữ liệu có thể thực hiện những việc sau:

- Báo cáo không có lỗi đối với công việc đến hoặc công việc hiện có trong quá trình bảo trì
- Chuyển hướng các phiên cơ sở dữ liệu đang hoạt động sang các dịch vụ chức năng khác
- Cân bằng lại các phiên cơ sở dữ liệu, nếu cần, trong và sau khi bảo trì

Kiểm soát hành vi thoát nước trong quá trình bảo trì theo kế hoạch bằng cách sử dụng các thuộc tính dịch vụ `Drain_timeout` và `stop_option` của tiện ích `SRVCTL`, Tiện ích kiểm soát dịch vụ dữ liệu toàn cầu (`GDSCCTL`) và Nhà môi giới bảo vệ dữ liệu Oracle. Gói `DBMS_SERVICE` cung cấp cơ sở hạ tầng cơ bản.



#### Xem thêm:

- Hướng dẫn triển khai và quản trị cụm ứng dụng thực của Oracle để tìm hiểu thêm
- Tham khảo các loại và gói PL/SQL của cơ sở dữ liệu Oracle để tìm hiểu thêm về `DBMS_SERVICE`
- Hướng dẫn triển khai và quản trị cụm ứng dụng thực của Oracle để tham khảo lệnh `SRVCTL`
- Khái niệm và quản trị dịch vụ dữ liệu toàn cầu của cơ sở dữ liệu Oracle Hướng dẫn tham khảo lệnh `GDSCCTL`

## Kiến trúc liên tục của ứng dụng

Các thành phần chính của Tính liên tục của ứng dụng là thời gian chạy, kết nối lại và phát lại.

Các giai đoạn như sau:

### 1. Thời gian chạy bình thường

Trong giai đoạn này, Tính liên tục của ứng dụng thực hiện các tác vụ sau:

- Xác định các yêu cầu cơ sở dữ liệu
- Quyết định xem các cuộc gọi cục bộ và cơ sở dữ liệu có thể được thực hiện lại hay không
- Xây dựng các đối tượng proxy để cho phép phát lại, nếu cần và quản lý hàng đợi
- Giữ các cuộc gọi ban đầu và xác thực các cuộc gọi này cho đến khi kết thúc cơ sở dữ liệu yêu cầu hoặc phát lại bị vô hiệu hóa

### 2. Kết nối lại

Giai đoạn này được kích hoạt bởi một lỗi có thể phục hồi. Ứng dụng liên tục thực hiện các nhiệm vụ sau:

- Đảm bảo tính năng phát lại được bật cho các yêu cầu cơ sở dữ liệu
- Quản lý thời gian chờ
- Có được kết nối mới tới cơ sở dữ liệu và sau đó xác thực rằng đây là mục tiêu cơ sở dữ liệu hợp lệ
- Sử dụng Transaction Guard để xác định xem giao dịch cuối cùng đã được thực hiện hay chưa thành công (giao dịch đã cam kết không được gửi lại)

### 3. Phát lại

Ứng dụng liên tục thực hiện các nhiệm vụ sau:

- Phát lại các cuộc gọi được giữ trong hàng đợi
- Tắt tính năng phát lại nếu những thay đổi mà người dùng có thể nhìn thấy trong kết quả xuất hiện trong quá trình phát lại
- Không cho phép thực hiện cam kết, nhưng cho phép lần thu hồi cuối cùng (đã gặp lỗi) thực hiện cam kết

Sau khi phát lại thành công, yêu cầu sẽ tiếp tục từ điểm thất bại.



Xem thêm:

- "Tổng quan về Bảo vệ giao dịch"
- Hướng dẫn triển khai và quản trị cụm ứng dụng thực của Oracle để tìm hiểu thêm về Tính liên tục của ứng dụng
- Hướng dẫn dành cho nhà phát triển cơ sở dữ liệu Oracle JDBC và Cơ sở dữ liệu Oracle JDBC  
Tham khảo API Java để tìm hiểu thêm về JDBC và Ứng dụng Liên tục

## Tổng quan về giao dịch tự chủ

Giao dịch tự trị là một giao dịch độc lập có thể được gọi từ một giao dịch khác, là giao dịch chính. Bạn có thể tạm dừng giao dịch gọi điện, thực hiện các thao tác SQL và cam kết hoặc hoàn tác chúng trong giao dịch tự động, sau đó tiếp tục giao dịch gọi điện.

Giao dịch tự động rất hữu ích cho các hành động phải được thực hiện độc lập, bất kể giao dịch gọi có cam kết hay quay lại hay không. Ví dụ: trong một giao dịch mua cổ phiếu, bạn muốn cam kết dữ liệu khách hàng bất kể giao dịch mua cổ phiếu tổng thể có diễn ra hay không. Ngoài ra, bạn muốn ghi thông báo lỗi vào bảng gỡ lỗi ngay cả khi toàn bộ giao dịch quay trở lại.

Giao dịch tự chủ có các đặc điểm sau:

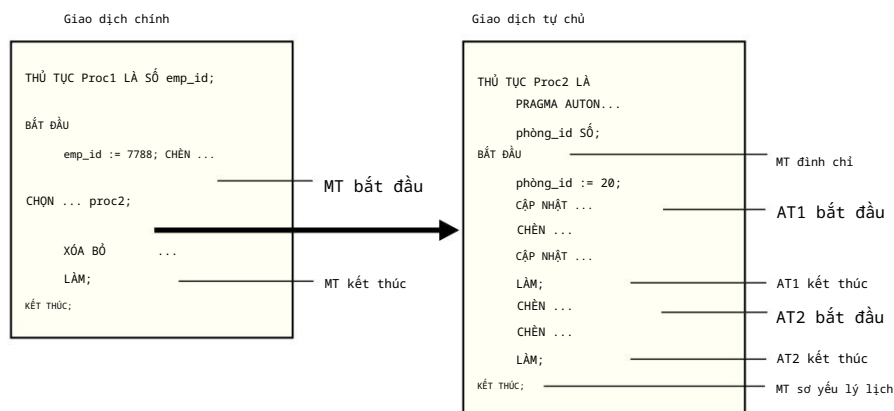
- Giao dịch tự trị không thấy những thay đổi không được cam kết được thực hiện bởi giao dịch chính và không chia sẻ khóa hoặc tài nguyên với giao dịch chính.
- Những thay đổi trong giao dịch tự trị được hiển thị đối với các giao dịch khác khi cam kết giao dịch tự trị. Nhờ đó, người dùng có thể truy cập thông tin cập nhật mà không cần phải đợi giao dịch chính được thực hiện.
- Các giao dịch tự chủ có thể bắt đầu các giao dịch tự chủ khác. Không có các giới hạn, ngoài giới hạn tài nguyên, về số lượng giao dịch tự trị có thể được gọi.

Trong PL/SQL, một giao dịch tự trị thực hiện trong phạm vi tự trị, đây là một quy trình được đánh dấu bằng pragma AUTONOMOUS\_TRANSACTION. Trong ngữ cảnh này, các quy trình bao gồm các khối PL/SQL ẩn danh cấp cao nhất cũng như các chương trình con và trình kích hoạt PL/SQL. Pragma là một [lệnh](#) hướng dẫn trình biên dịch thực hiện tùy chọn biên dịch. Pragma AUTONOMOUS\_TRANSACTION hướng dẫn cơ sở dữ liệu rằng thủ tục này, khi được thực thi, sẽ được thực thi như một giao dịch tự trị mới độc lập với giao dịch gốc của nó.



Đồ họa sau đây cho thấy cách điều khiển chuyển từ quy trình chính (MT) sang quy trình tự trị và ngược lại. Thủ tục chính là Proc1 và thủ tục tự trị là Proc2. Quy trình tự trị có thể thực hiện nhiều giao dịch (AT1 và AT2) trước khi điều khiển quay trở lại quy trình chính.

Hình 10-4 Luồng kiểm soát giao dịch



Khi bạn vào phần thực thi của một quy trình tự trị, quy trình chính sẽ tạm dừng. Khi bạn thoát khỏi quy trình tự động, quy trình chính sẽ tiếp tục lại.

Trong [Hình 10-4](#), COMMIT bên trong Proc1 tạo ra vĩnh viễn không chỉ công việc của chính nó mà còn bất kỳ công việc nổi bật nào được thực hiện trong phiên của nó. Tuy nhiên, CAM KẾT trong proc2 chỉ khiến công việc được thực hiện trong giao dịch proc2 trở thành vĩnh viễn. Do đó, câu lệnh COMMIT trong giao dịch AT1 và AT2 không có tác dụng gì đối với giao dịch MT.



Xem thêm:

Hướng dẫn phát triển cơ sở dữ liệu Oracle và cơ sở dữ liệu Oracle PL/SQL  
Tham khảo ngôn ngữ để tìm hiểu cách sử dụng các giao dịch tự trị

## Tổng quan về giao dịch phân phối

Giao dịch phân tán là giao dịch bao gồm một hoặc nhiều câu lệnh cập nhật dữ liệu trên hai hoặc nhiều nút riêng biệt của cơ sở dữ liệu phân tán, sử dụng đối tượng lược đồ được gọi là liên kết cơ sở dữ liệu.

**Cơ sở dữ liệu phân tán** là một tập hợp cơ sở dữ liệu trong hệ thống phân tán có thể xuất hiện trong các ứng dụng dưới dạng một nguồn dữ liệu duy nhất. **Liên kết cơ sở dữ liệu** mô tả cách một phiên bản cơ sở dữ liệu có thể đăng nhập vào một phiên bản cơ sở dữ liệu khác.

Không giống như giao dịch trên cơ sở dữ liệu cục bộ, giao dịch phân tán sẽ thay đổi dữ liệu trên nhiều cơ sở dữ liệu. Do đó, việc xử lý giao dịch phân tán phức tạp hơn vì cơ sở dữ liệu phải phối hợp thực hiện hoặc khôi phục các thay đổi trong giao dịch dưới dạng một đơn vị nguyên tử. Toàn bộ giao dịch phải cam kết hoặc quay trở lại. Cơ sở dữ liệu Oracle phải phối hợp kiểm soát giao dịch qua mạng và duy trì tính nhất quán của dữ liệu, ngay cả khi xảy ra lỗi mạng hoặc hệ thống.



### Xem thêm:

Hướng dẫn dành cho quản trị viên cơ sở dữ liệu Oracle để tìm hiểu cách quản lý các giao dịch phân tán

## Cam kết hai giai đoạn

Cơ chế cam kết hai giai đoạn đảm bảo rằng tất cả các cơ sở dữ liệu tham gia vào giao dịch phân tán đều cam kết hoặc hoàn tác tất cả các câu lệnh trong giao dịch.

Cơ chế này cũng bảo vệ DML tiềm ẩn được thực hiện bằng các ràng buộc toàn vẹn, các lệnh gọi thủ tục từ xa và trình kích hoạt.

Trong cam kết hai giai đoạn giữa nhiều cơ sở dữ liệu, một cơ sở dữ liệu sẽ điều phối giao dịch phân tán.

Nút khởi tạo được gọi là điều phối viên toàn cầu. Điều phối viên hỏi các cơ sở dữ liệu khác xem họ có sẵn sàng cam kết hay không. Nếu bất kỳ cơ sở dữ liệu nào phản hồi là không thì toàn bộ giao dịch sẽ được khôi phục. Nếu tất cả các cơ sở dữ liệu bỏ phiếu đồng ý thì điều phối viên sẽ phát đi một thông báo để thực hiện cam kết vĩnh viễn trên mỗi cơ sở dữ liệu.

Cơ chế cam kết hai giai đoạn minh bạch đối với người dùng thực hiện các giao dịch phân tán. Trên thực tế, người dùng thậm chí không cần biết giao dịch được phân phối. Câu lệnh COMMIT biểu thị sự kết thúc của giao dịch sẽ tự động kích hoạt cơ chế cam kết hai pha. Không cần mã hóa hoặc cú pháp câu lệnh phức tạp để bao gồm các giao dịch phân tán trong phần thân của ứng dụng cơ sở dữ liệu.



### Xem thêm:

- Hướng dẫn dành cho quản trị viên cơ sở dữ liệu Oracle để tìm hiểu về cơ chế cam kết hai giai đoạn
- Tham khảo ngôn ngữ SQL của cơ sở dữ liệu Oracle

## Giao dịch đáng ngờ

Giao dịch phân tán không rõ ràng xảy ra khi cam kết hai giai đoạn bị gián đoạn do bất kỳ loại lỗi hệ thống hoặc mạng nào.

Ví dụ: hai cơ sở dữ liệu báo cáo với cơ sở dữ liệu phối hợp rằng chúng đã được chuẩn bị để cam kết, nhưng phiên bản cơ sở dữ liệu phối hợp không thành công ngay sau khi nhận được thông báo. Hai cơ sở dữ liệu đã sẵn sàng cam kết hiện đang bị treo trong khi chờ thông báo kết quả.

Quá trình nền của trình khôi phục (RECO) tự động giải quyết kết quả của các giao dịch phân tán nghi ngờ.

Sau khi lỗi được sửa chữa và liên lạc được thiết lập lại, quy trình RECO của từng cơ sở dữ liệu Oracle cục bộ sẽ tự động cam kết hoặc khôi phục mọi giao dịch phân tán đáng nghi ngờ một cách nhất quán trên tất cả các nút liên quan.

Trong trường hợp xảy ra lỗi lâu dài, Cơ sở dữ liệu Oracle cho phép mỗi quản trị viên cục bộ cam kết hoặc hoàn tác thủ công mọi giao dịch phân tán bị nghi ngờ do lỗi. Tùy chọn này cho phép người quản trị cơ sở dữ liệu cục bộ giải phóng mọi tài nguyên bị khóa bị giữ vô thời hạn do lỗi lâu dài.

Nếu cơ sở dữ liệu phải được khôi phục về một thời điểm trong quá khứ thì các phương tiện khôi phục cơ sở dữ liệu sẽ cho phép quản trị viên cơ sở dữ liệu tại các địa điểm khác đưa cơ sở dữ liệu của họ về thời điểm trước đó. Hoạt động này đảm bảo rằng cơ sở dữ liệu toàn cầu vẫn nhất quán.



#### Xem thêm:

- "Quy trình phục hồi (RECO) "
- Hướng dẫn dành cho quản trị viên cơ sở dữ liệu Oracle để tìm hiểu cách quản lý tình trạng nghi ngờ giao dịch