**Bài tập thực hành buổi 3 (Tại lớp)**

**Mô tả thao thác của Trigger dẫn đến Mutating-Table Error.**

Ví dụ về mô tả thao tác của trigger TRG\_EMPLOYEES dẫn đến lỗi "Mutating-Table Error":

* Bảng EMPLOYEES có một trigger là TRG\_EMPLOYEES được kích hoạt sau mỗi lần cập nhật.
* Khi một hàng trong bảng EMPLOYEES được cập nhật, trigger bắt đầu thực hiện.
* Trigger cố gắng truy vấn bảng EMPLOYEES để tính toán mức lương trung bình của tất cả nhân viên.
* Tuy nhiên, trạng thái của bảng EMPLOYEES không nhất quán do bản cập nhật chưa được xác nhận.
* Do đó, khi trigger cố gắng truy cập bảng EMPLOYEES, lỗi "Mutating-Table Error" xảy ra, vì bảng đang trong quá trình thay đổi và không thể truy cập cùng một lúc.

**Compound trigger: định nghĩa, đặc điểm và ứng dụng.**

Định nghĩa: là một trigger duy nhất trên một bảng cho phép bạn chỉ định các hành động cho mỗi trong bốn điểm thời gian sau đây:

* Trước câu lệnh kích hoạt
* Trước mỗi hàng mà câu lệnh kích hoạt ảnh hưởng đến
* Sau mỗi hàng mà câu lệnh kích hoạt ảnh hưởng đến
* Sau câu lệnh kích hoạt.

Đặc điểm của Compound Trigger:

* Một cấu trúc duy nhất: Compound trigger cho phép kết hợp nhiều loại trigger và hành động vào một cấu trúc duy nhất, bao gồm BEFORE STATEMENT, BEFORE ROW, AFTER ROW và AFTER STATEMENT. Điều này giúp quản lý logic của trigger dễ dàng hơn và giảm thiểu số lượng trigger cần tạo ra.
* Hỗ trợ cho nhiều hoạt động: Compound trigger cung cấp khả năng thực hiện nhiều hoạt động khác nhau trong một lần kích hoạt trigger, bao gồm truy vấn, kiểm tra điều kiện và thay đổi dữ liệu.
* Kiểm soát kết quả của giao dịch: Compound trigger cho phép kiểm soát kết quả của giao dịch bằng cách sử dụng các phần BEFORE STATEMENT và AFTER STATEMENT để kiểm soát hành vi của giao dịch và thực hiện các hành động phù hợp.

Ứng dụng của Compound Trigger:

* Compound trigger để giải quyết lỗi Mutating-Table Error. Với compound trigger, có thể chia nhỏ logic của trigger thành nhiều phần, bao gồm trước và sau khi thực hiện thay đổi trên các hàng của bảng, cũng như trước và sau khi thực hiện thay đổi trên toàn bộ câu lệnh. Bằng cách này, có thể kiểm soát được thứ tự thực thi của các phần logic và tránh được lỗi "mutating table error".
* Auditing (Kiểm tra việc sử dụng dữ liệu): Compound trigger được sử dụng để ghi lại các hoạt động của người dùng đối với dữ liệu trong hệ thống, bao gồm việc ghi log (logging) các thao tác INSERT, UPDATE và DELETE.
* Quản lý tính toàn vẹn dữ liệu (Data Integrity Management): Compound trigger có thể được sử dụng để thực hiện kiểm tra và điều chỉnh dữ liệu để đảm bảo tính toàn vẹn của dữ liệu trong cơ sở dữ liệu.
* Thực hiện logic phức tạp (Complex Business Logic): Compound trigger cung cấp khả năng thực hiện logic phức tạp và tùy chỉnh để đáp ứng các yêu cầu kinh doanh cụ thể của ứng dụng.

**AUTONOMOUS\_TRANSACTION: định nghĩa, đặc điểm và ví dụ.**

Autonomous transaction là giao dịch độc lập được bắt đầu bởi một giao dịch khác, giao dịch chính. Giao dịch tự động cho phép bạn tạm dừng giao dịch chính, thực hiện các thao tác SQL, cam kết hoặc khôi phục các hoạt động đó, sau đó tiếp tục giao dịch chính.

Đặc điểm của autonomous transaction:

* Giao dịch Độc lập: autonomous transaction cho phép thực hiện các thao tác ghi mà không phụ thuộc vào kết quả của giao dịch chính đang chạy. Điều này giúp tránh tình trạng chặn (blocking) và giảm thiểu nguy cơ gặp lỗi "mutating table error".
* Không chờ đợi giao dịch chính: autonomous transaction không chờ đợi giao dịch chính hoàn tất trước khi bắt đầu thực hiện. Nó được thực hiện độc lập và không bị ảnh hưởng bởi trạng thái của giao dịch chính.
* Làm giảm lỗi "mutating table error": autonomous transaction thường được sử dụng để giải quyết lỗi "mutating table error" bằng cách thực hiện các thao tác ghi trên bảng mà chính trigger đang kích hoạt trên đó.

Ví dụ:

CREATE OR REPLACE TRIGGER trg\_example

AFTER INSERT ON table\_name

FOR EACH ROW

DECLARE

PRAGMA AUTONOMOUS\_TRANSACTION; -- Khai báo giao dịch độc lập

BEGIN

-- Thực hiện các thao tác ghi không phụ thuộc vào giao dịch chính

INSERT INTO log\_table (message) VALUES ('New row inserted');

COMMIT; -- Kết thúc giao dịch độc lập

END;

/

**Mô tả kết quả và giải thích các trường hợp trong Mục 2.3. Tìm hiểu và cài đặt mutating trigger và compound trigger.**

**Trường hợp 1:**

**A close-up of a message

Description automatically generated**

Trigger TRG\_TEST cố gắng thực hiện truy vấn SELECT trên bảng employee trong khi trigger đang được kích hoạt sau mỗi lần cập nhật cột salary của bảng employee. Điều này gây ra lỗi "mutating table", vì bảng employee đang trong quá trình thay đổi và không thể truy cập hoặc thay đổi dữ liệu trong khi trigger đang thực hiện.

**Trường hợp 2:**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Trong trigger TRG\_TEST, sau khi cập nhật cột salary trên bảng employee, trigger thực hiện một truy vấn SELECT để đếm số nhân viên có mức lương là 1000 và hiển thị kết quả ra output. Khi cập nhật được thực hiện, có 2 nhân viên đã có mức lương được thay đổi từ 800 thành 1000, do đó kết quả là "Total employee are 2".

**Trường hợp 3:**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Trong trigger TRG\_TEST, sau mỗi lần cập nhật cột salary trên bảng employee, trigger thực hiện một giao dịch độc lập (AUTONOMOUS\_TRANSACTION) để đếm số nhân viên có mức lương là 1000 và hiển thị kết quả ra output. Khi cập nhật được thực hiện, không có nhân viên nào có mức lương được thay đổi từ 800 thành 1000, do đó kết quả là "Total employee are 0".

**Trường hợp 4:**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Trigger TRG\_TEST được kích hoạt sau mỗi lần cập nhật trên bảng employee. Sau mỗi hàng được cập nhật, trigger hiển thị thông điệp "Update is done". Sau khi tất cả các cập nhật đã hoàn thành, trigger đếm số nhân viên có mức lương là 1000 và hiển thị kết quả.