1. 데이터베이스 트리거

테이블에 어떤 조작이 가해졌을 때에 미리 지정해 놓은 처리를 자동으로 실행시키는 블록을 말하며, PL/SQL 블록으로 작성하며 오라클 데이터베이스에 저장된다.

데이터베이스 트리거는 테이블, 뷰에 대한 처리 내용, 실행 조건, 실행 시간 등을 설정하고, 설정 조건에 따라 자동으로 실행된다. 특히 데이터가 변경되는 중용한 테이블에 트리거를 설정하면 문제가 발생할 경우 데이터 추적이 가능하고, 너무 많이 사용하면 성능이 저하되는 문제점도 발생한다.

데이터베이스 트리거는 스키마나 사용자, 테이블, 뷰 등의 수준에서 정의되고, 이벤트가 발생하면 실행된다. 이벤트가 발생되는 경우는 다음과 같다.

- 데이터베이스 조작 DML(INSERT, UPDATE, DELETE)문 실행
- 데이터베이스 정의 DDL(CREATE, ALTER, DROP)문 실행
- 데이터베이스 동작 (LOGON, LOGOFF, STARTUP, SHUTDOWN, SERVERERROR) 실행

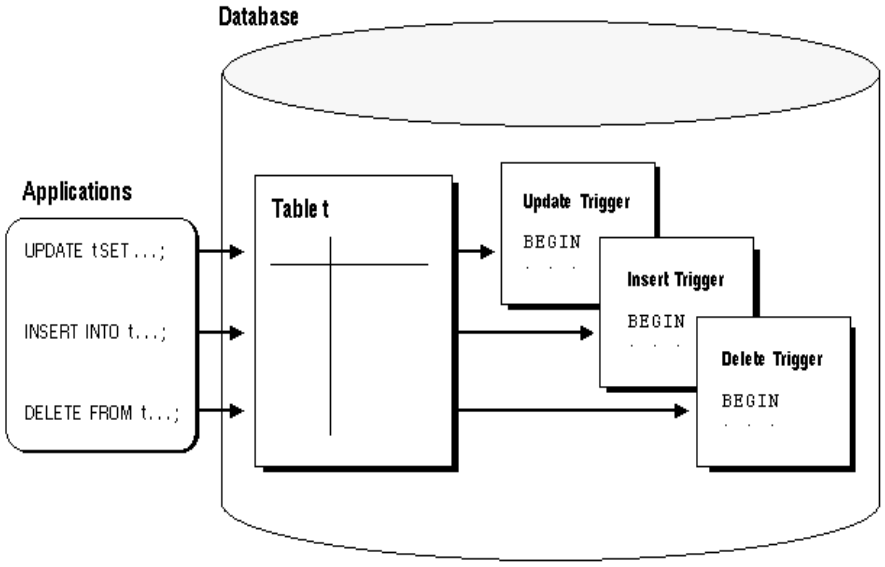
☞ **데이타베이스 트리거 용도**

- 고급의 보안 정책을 준수
- 데이터 무결성 유지 또는 방지
- 기본 키를 비롯한 칼럼 값 자동 생성
- 테이블의 변경 내용을 기록하여 관리
- 업무 규칙 이해
- 조건에 따라 DML문의 실행 허용 등

테이블이나 뷰 등에 정의하는 트리거를 데이터베이스 트리거 또는 트리거(trigger)라고도 한다. DML 트리거와 INSTEAD OF 트리거, non-DML 트리거에 대해서 작성한다.

2. DML 트리거

INSERT, UPDATE, DELETE문에 의해 테이블의 내용이 변경될 때마다 자동으로 실행되는 PL/SQL블록을 말한다. DML 트리거는 사용자가 테이블에 이벤트가 발생할 때마다 자동적으로 실행되며, PL/SQL 블록으로 작성하고, 오라클 데이터베이스에 저장된다.



2.1. DML 트리거 생성 구문

DML 트리거는 테이블이나 뷰에 대해서 정의하는 것으로

- 테이블에 행이 추가되는 이벤트(INSERT문 실행)
- 테이블에 행의 칼럼 값이 수정되는 이벤트(UPDATE문 실행)
- 테이블에 행이 삭제되는 이벤트(DELETE문 실행)이며,

특정 테이블에 대하여 어떤 이벤트가 발생할 때 언제, 어떻게 처리하는가를 정의하며, 트리거를 생성하는 구문은 다음과 같다.

문법	CREATE [OR REPLACE] TRIGGER 트리거명 [BEOFRE | AFTER] Triggering-event ON 테이블명 [FOR EACH ROW] [WHEN 조건] PL/SQL block;

※ 기술 방법

• triggering-event : DML의 INSERT, UPDATE, DELETE을 기술한다. UPDATE문은 [UPDATE OF 칼럼명1, 칼럼명2, ····]로 기술할 수 있고, 이때 지정된 칼럼명에 대해서만 이벤트가 발생하며, 생략하면 모든 칼럼에 대하여 이벤트가 발생한다.

• 테이블명 : 데이터베이스 트리거를 적용할 테이블명을 기술한다.

• BEFORE | AFTER : BEFORE는 트리거링 이벤트(triggering-event) 발생전에 PL/SQL block을 실행하고, AFTER는 트리거링 이벤트 발생 후에 PL/SQL block을 실행한다.

• FOR EACH ROW : 선택이며 각 행이 변환 때마다 실행되는 행 수준의 트리거일 때 지정한다.

• WHEN 조건 : 트리거가 실행되는 조건을 기술한다.

• PL/SQL block : 트리거에서 처리할 블록(block)의 명령문을 기술한다.

2.2 DML 트리거의 유형

DML 트리거는 문장 수준과 행 수준의 트리거, DML 문장, 타이밍에 따라 여러 유형이 있다.

☞ 문장 수준과 행 수준의 트리거

문장 수준의 트리거는 한 문장의 실행에 의한 변경 또는 삭제되는 행의 수와 관계없이 한번만 이벤트가 발생되고 실행된다.

- FOR EACH ROW 구문을 포함하지 않는다.
- 행이 추가되거나 수정되거나 삭제될 때마다 실행한다.

☞ DML 문장

- INSERT는 행이 추가될 때 실행된다.
- UPDATE는 모든 칼럼이 변경될 때 실행되고, UPDATE OF 칼럼명1, ····은 지정한 칼럼에 대한 변경이 될 때 실행된다.
- DELETE는 행이 삭제될 때 실행된다.
- INSERT OR UPDATE OR DELETE, MERGE는 행의 추가나 칼럼 변경, 행의 삭제시 실행된다.

☞ 타이밍(timing)

- BEFORE는 테이블에 DML문장이 실행되기 전에 트리거가 실행된다.
- AFTER는 테이블에 DML 문장이 실행한 후 트리거를 실행한다.

따라서 테이블에 적용 가능한 트리거의 유형은

- 트리거링 이벤트(triggering-event)에 기술할 명령문의 3 종류
- FOR EACH ROW의 기술 유무에 의한 문장 수준과 행 수준 트리거 2 종류
- 트리거 실행 시기에 관한 BEFORE, AFTER 의 2 종류

을 종합하면, DML 트리거는 3*2*2=12가지 유형이 된다.

2.3. DML 트리거에서 칼럼 값 참조

DML 트리거의 PL/SQL 블록에서 테이블에 입력, 수정, 삭제될 때 테이블에 관련된 값을 참조할 수 있다. 이 값은 DML 트리거가 실행될 때 :new, :old 두 종류의 의사 레코드(Pseudo Record)를 통하여 DML 트리거에 나타난다.

☞ 의사 레코드 형식

문법	:new. 칼럼명 :old. 칼렴명

※ 여기서 칼럼명은 DML 트리거에 정의된 테이블의 칼럼명이다.

즉, 다음의 명령문이 실행될 때, DML 트리거의 PL/SQL 블록에는

- INSERT문이면 추가할 행의 칼럼 값이 ":new.칼럼명"으로 나타난다.
- UPDATE문이면 수정할 행의 수정전 칼럼 값이 ":old.칼럼명"에 나타나고, 수정할 칼럼 값이 ":new.칼럼명"에 나타난다.
- DELETE문이면 삭제할 행의 칼럼 값이 ":old.칼럼명"에 나타난다.

테이블에 저장되거나 수정되거나 삭제될 때 트리거 본문에서 의사 레코드를 이용하여 참조할 수 있다.

☞ DML 트리거의 술어

DML 트리거에서 테이블의 행에 대한 추가나, 수정, 삭제에 관한 변경 데이터를 저장하기 위해서, DML문의 어떤 문장 수준인지를 확인하는 INSERTING, UPDATING, DELETING 세 가지 술어가 있다.

- INSERTING은 트리거링 문장이 INSERT문이면 참이 되고, 그렇지 않으면 거짓이 된다.
- UPDATING은 트리거링 문장이 UPDATE문이면 참이 되고, 그렇지 않으면 거짓이 된다.
- DELETING은 트리거링 문장이 DELTE문이면 참이 되고, 그렇지 않으면 거짓이 된다.

3. DML 트리거 작성

3.1. DML 트리거 고려사항

- DML 트리거의 캐스케이드(cascade)는 총 32개까지이다.
- DML 트리거에서 저장된 프로시저나 저장된 함수 등을 호출할 수 있다.
- DML 트리거에는 트랜잭션 제어문을 사용할 수 없다. 왜냐하면, 예외가 발생하거나, 변경된 데이터를 취소할 경우가 발생되었을 때, COMMIT문으로 저장된 값들은 ROLLBACK문으로 취소할 수 없기 때문이다.

3.2. DML 트리거에 사용할 테이블 생성

특정 테이블에 대한 DML 트리거를 작성하기 전에 동일한 구조를 갖는 테이블을 생성한다. 이 테이블은 서브 쿼리를 이용하면 쉽게 생성할 수 있다.

예제) SG_Scores 테이블의 DML 트리거에 사용할 테이블을 생성해보자.

3.3. DML 트리거 생성과 실행 확인

예제) SG_Scores 테이블에 INSERT문이 실행되기 전에 행 단위로 이벤트가 발생되는 SG_Scores_change_log 트리거를 생성하고, SG_Scores 테이블에 한 행을 추가하고, 트리거의 실행 결과를 확인해보자.

3.4. DML 트리거에서 칼럼 값 생성

테이블에 행을 추가, 수정할 때 DML 트리거에서 값을 생성할 수도 있고, DML 트리거의 PL/SQL 블록에서 저장된 프로시저 서브프로그램이나 저장된 함수 서브프로그램을 호출하여 실행할 수도 있다.

예제) 성적의 등급을 산출하는 Grade_Cal 함수를 만들고 이 함수를 호출하여 등급을 산출하는 Grade_Search 트리거를 만들어서 SG_Scores 테이블에 행을 추가하여 확인해보자.

4. DML 트리거 관리

생성된 DML 트리거의 목록 조회, 트리거의 활성화와 비활성화, 트리거 삭제를 할 수 있다.

4.1. 트리거 목록 조회

트리거의 목록 조회는 All_Triggers 뷰로부터 트리거명, 트리거링 이벤트, 트리거 타입, 테이블명 등의 칼럼으로 조회할 수 있다.

주요 칼럼명	설 명
Trigger_Name	트리거의 이름을 반환
Trigger_Owner	트리거를 생성한 오라클 계정을 반환
Triggering_Event	트리거의 트리거링 이벤트를 반환
Trigger_Type	트리거가 문장 또는 행 수준인가를 반환
Table_Name	트리거가 어느 객체에 지정되었는지 반환
Owner	검색할 오라클 계정을 기술

예제) 'DBTEST' 계정에 속한 트리거의 목록을 모두 출력해보자.

4.2. 트리거 활성화와 비활성화

ALTER TRIGGER 문으로 트리거의 실행을 비활성화하거나 비활성화된 트리거를 활성화시킬 수 있다.

☞데이터베이스 트리거 실행 중지

문법	ALTER TRIGGER 트리거명 DISABLE;

☞데이터베이스 트리거 활성화

문법	ALTER TRIGGER 트리거명 ENABLE;

예제) Grade_Search 트리거를 비활성화하고, SG_Scores 테이블에 행을 추가하여 트리거의 실행 여부를 확인해보자. 그리고 또 다시 활성화한 후 다시 행을 추가해서 트리거의 실행 여부를 확인해보자.

4.3. 트리거 삭제

생성된 트리거는 DROP TRIGGER문으로 삭제할 수 있다.

문법	DROP TRIGGER 트리거명;

예제) Grade_Search 트리거를 삭제하고, 삭제되었는지 확인해보자.

4.4. DML 트리거 제한 사항

• COMMIT, ROLLBACK, SAVEPOINT의 트랜잭션 제어문은 사용할 수 없다. 만약, 테이블의 변경 내용을 취소(rollback)할 경우, 데이터베이스 트리거에서 COMMIT하면 모든 변경 내용을 취소할 수 없기 때문이다.

• CREATE TABLE문 등과 같은 데이터 정의어는 데이터베이스 트리거 내에서 실행될 수 없고, 호출되는 프로시저나 함수에서도 실행될 수 없다.

• 현재 변경되고 있는 테이블을 데이터베이스 트리거에서 질의하거나, 변경하지 못한다.

5. INSTEAD OF 트리거

INSTEAD OF 트리거란 DML 명령문으로 직접 데이터를 변경할 수 없는 뷰를 변경할 때 사용하는 트리거를 말한다.

INSTEAD OF 트리거는 뷰에 대해서 정의하는 것으로,

- 뷰에 행이 추가되는 이벤트(INSERT문 실행)
- 뷰에 행의 칼럼 값이 수정되는 이벤트(UPDATE문 실행)
- 뷰에 행이 삭제되는 이벤트(DELETE문 실행)이며,

테이블 변경이 이루어지기 전에 작동하는 트리거이다. INSTEAD OF 트리거가 실행하면 INSERT, UPDATE, DELETE문은 무시된다.

문법	CREATE [OR REPLACE] TRIGGER 트리거명 Triggering-event ON 뷰명 [FOR EACH ROW] ....

※ 기술 방법

• triggering-event : DML의 INSTEAD OF INSERT, INSTEAD OF UPDATE, INSTEAD OF DELETE을 기술한다.

• 뷰명 : 트리거를 적용할 뷰명을 기술한다.

예제) SG_Scores 테이블에 대한 접근성을 제한하기 위하여 구조가 동일한 SG_Scores_View 뷰를 만들고, SG_Scores_View 뷰에 INSTEAD문이 실행되기 전에 행 단위로 이벤트가 발생되는 SG_Scores_View_log 트리거를 만들어보자. 그리고 SG_Scores_View 뷰에 행을 추가하여 확인하자.