



PL/SQL Trigger

이러한 의미처럼 트리거(TRIGGER)란 DML(Data Manipulation Language) 작업 즉, [INSERT, UPDATE, DELETE] 작업이 일어날때 **자동적으로 실행되는(유발되는, 촉발되는) 객체**

이다. (이와 같은 특징을 강조하여 DML TRIGGER 라고 부르기도 한다)

트리거의 분류

분류에 따른 트리거의 종류

trigger는 활성화 및 비활성화가 가

변경되기 전에값 및 업데이트전 값 선택

DDL 트리거 DDL 트리거는 **CREATE, ALTER, DROP** 작업을 할 때 작동 된다.

DB변경시 잡아내는 Trigger

server 변경시 잡아내는 trigger

DDL Trigger 활성화 및 비활성화

트리거의 분류

SQL문 실행시기에 따른 분류	before 트리거	INSERT,UPDATE,DELETE 실행하기 전에 트리거 먼저 실행 됨
after 트리거	INSERT,UPDATE,DELETE 실행하고 난 후 트리거 실행 됨	
실행시 영향을 받는 곳에 따른 분류	row 트리거 (행 트리거)	실행된 트리거가 row(행) 하나하나마다 실행 됨
statement 트리거(문장 트리거)	실행된 트리거가 INSERT,UPDATE,DELETE 문장에 1번만 실행 됨	

분류에 따른 트리거의 종류

밑에 순서대로 실행순서를 갖는다.

실행시점	실행범위	내용
------	------	----

before	statement	SQL문을 실행하기 전에 그 문장에 대해 한번 실행
before	row	SQL문을 실행하기 전에 각각 row(행)에 대해 한번씩 실행
after	statement	SQL문을 실행하고 난 후 그 문장에 대해 한번 실행
after	row	SQL문을 실행하고 난 후 각각 row(행)에 대해 한번씩 실행

```
CREATE [OR REPLACE] TRIGGER 트리거명
```

```
BEFORE or AFTER
```

```
트리거이벤트 ON 테이블명
```

```
[FOR EACH ROW [WHEN TRIGGER 조건]]
```

```
[Referencing OLD AS {변경전 값을 참조하는 변수명} NEW AS {변경후 값을  
DECLARE
```

```
선언문
```

```
BEGIN
```

```
PL/SQL 코드
```

```
END;
```

```
-----
```

```
CREATE OR REPLACE TRIGGER welcome
```

```
    AFTER INSERT ON emp2 --emp2테이블에서 행입력이 발생했다면
```

```
    FOR EACH ROW -- update 같은 여러개 행에 적용가능 (각행에 적)
```

```
    BEGIN
```

```
        DBMS_OUTPUT.PUT_LINE('환영합니다');
```

```
    END;
```

```
    /
```

```
INSERT INTO emp2 VALUES (7); -- INSERT시 트리거 실행
```

trigger는 활성화 및 비활성화가 가

```
--1.  
ALTER TRIGGER TEST_TRIGGER ENABLE;  
ALTER TRIGGER TEST_TRIGGER DISABLE;  
--2.  
ALTER TABLE TEST_TABLE DISABLE ALL TRIGGERS;  
ALTER TABLE TEST_TABLE DISABLE ALL TRIGGERS;
```

변경되기 전에값 및 업데이트전 값 선택

old : UPDATE 되기전의 값, DELETE 되기전의 값에 접근 가능

new : INSERT 되는 값, UPDATE된 후의 값에 접근 가능

DDL 트리거 DDL 트리거는 CREATE, ALTER, DROP 작업을 할 때 작동 된다.

```
CREATE OR REPLACE TRIGGER before_create_trigger  
after DDL  
ON SCHEMA  
BEGIN  
    INSERT INTO ddl_log --해당테이블에 변경내역 저장 (실제있는 테이블)  
    SELECT ora_sysevent, ora_dict_obj_owner,  
           ora_dict_obj_name, USER, SYSDATE  
    FROM DUAL;  
END;
```

DB변경시 잡아내는 Trigger

```
ALTER TRIGGER ddlTrg_ProhibitSonDB
ON DATABASE
AFTER DDL_TABLE_VIEW_EVENTS
AS
    PRINT N'경고 : 모든 테이블/뷰/인덱스/통계에 대한 생성/변경/삭제 금지'
    ROLLBACK TRAN
GO
```

server 변경시 잡아내는 trigger

```
CREATE TRIGGER ddlTrg_Server
ON ALL SERVER
AFTER CREATE_DATABASE, ALTER_DATABASE, DROP_DATABASE
-- 이는 AFTER DDL_DATABASE_EVENTS 그룹과 동일하다
AS
    PRINT N'경고 : 데이터베이스 관련된 작업은 관리자에게 문의하세요'
    ROLLBACK TRAN
GO
```

DDL Trigger 활성화 및 비활성화

```
-- 비활성화
DISABLE TRIGGER ddlTrg_ProhibitSonDB ON DATABASE
GO

-- 활성화
ENABLE TRIGGER ddlTrg_ProhibitSonDB ON DATABASE
GO

-- 서버 범위
DISABLE TRIGGER ddlTrg_Server ON ALL SERVER
GO
```

```
CREATE DATABASE prohibitedDB  
GO
```