REM **TRIGGER**

1. What**?**
   1)사전적 의미는 **방**아쇠
   2)**방**아쇠**를** 당기는 이벤트가 발생하면 총알이 발사된다는 의미
   3)오**라**클에서 어떤 이벤트가 발생할 경우 연관된 다른 작업이 자동**으**로 수행된다는 **뜻**
   4)예
      -입고 테이블에 상품이 입고되면 재고 테이블에 자동**으**로 재고가 증가하게만드는것
      -데이타베이스**를** 사용하다 보면 원치 않는 데이터가 사용자들에 의해 변경되는 **일**이 자주 발생한다**.**
      -또한**,** 데이터베이스 사용이 인증되지 않은 사용자가 임의로 테이블을 변경하는 **일**이 발생하기도 한다**.**
      -만**약,**데이터베이스 사용자 계정과 암호**를** 알고 있다면 사용자의 이런 변경 작업을 **방**지할 수 있는 **방**법은 전혀 없게 된다**.**
      -이럴 경우**,** 데이터베이스 트리거**를** 이용하면 원치않는 시간에 원치 않는 사용자가 원치 않는 테이블에 변경 작업하는 것을 **방**지할 수 있다**.**
   5)트리거는 데이터베이스 내에 오브젝트로서 저장되어 관리된다**.**
   6)트리거 자체는 사용자가 지정해서 실행할 수 없**으**며**,** 오직 트리거 생성 시 정의한 특정 사건**(**이벤트**)**에 의해서만 묵시적인 자동실행**(**Fire**)**이 이루어진다**.**
   7)**TRIGGER** 몸체**(**실행부**)**에는 TCL 명령**,** 즉 **COMMIT, ROLLBACK, SAVEPOINT** 명령이 포함될 수 없다**.**

2. 권한
   1)생성하려면 **CREATE TRIGGER** 권한이 필요
   2)수정하려면 **ALTER TRIGGER** 권한이 필요
   3)삭제하려면 **DROP TRIGGER** 권한이 필요
   4)데이터베이스 전체의 **TRIGGER** 조작은 administer DATABASE **TRIGGER** 시스템 권한이 필요

3. 조회
   1)USER_OBJECTS
   2)USER_TRIGGERS
   3)USER_ERRORS

4. **TRIGGER** 구성요소
   1)트리거 유형 **: STATEMENT LEVEL, ROW LEVEL :** 트리거 몸체의 내용이 몇 번이나 실행되는가**?**
   2)트리거 타이밍 **: BEFORE, AFTER :** 사용자가 트리거 이벤트**를** 유발시킬 때 데이터베이스 트리거**를** 언제 실행하는가**?**
   3)트리거 이벤트 **: INSERT, UPDATE, DELETE :** 테이블 상에 어떤 데이터 조작 연산이 트리거**를** 발생시키는가**?**
   4)트리거 몸체 **:** PL**/SQL** 블**록 :** 어떤 이벤트가 발생하면 어떤 작업이 수행되는가**?**
   5)트리거 조건 **: WHEN [**조건**] :** 사용자의 트리거 이벤트 중에 조건을 만족하는 데이터만 트리거 한다**.**

5. Syntax

```
CREATE [OR REPLACE ] TRIGGER trigger_name
{BEFORE | AFTER | INSTEAD OF } --타이밍
{INSERT [OR] | UPDATE [OR] | DELETE} --이벤트
[OF col_name]
ON {table_name | view_name | SCHEMA | DATABASE }
[REFERENCING OLD AS o NEW AS n]
[FOR EACH ROW] WHEN (condition)   --행 레벨 트리거
--trigger body
DECLARE
   Declaration-statements
BEGIN
   Executable-statements
```

```
EXCEPTION
   Exception-handling-statements
END;
```

-OR REPLACE : 생성하고자 하는 트리거가 기존에 동일명으로 존재할 경우, 기존의 내용을 현재의 내용으로 수정하는 옵션

-trigger_name : 같은 도메인내에서는 중복되어서는 안된다.

-timing : TRIGGER 가 실행되는 시점을 지정하는 것으로, 이벤트 발생 전과후를 의미하는 BEFORE 와 AFTER 가 있으며, TRIGGER 가 특정 VIEW 에 대한 DML 일 경우에는 timing 부분에 INSTEAD OF 를 사용한다.

-event : TRIGGER 를 실행시키는 사건으로 테이블/뷰에 관련된 DML 이벤트와 SCHEMA /DATABASE에 관련된 DDL 이벤트와 DATABASE 이벤트로 나뉜다.

-ON : TRIGGER 가 발생되는 레벨 또는 대상을 지정하는 절

-REFERENCING : 처리되는 각각의 행에대해 변경전의 값과 변경 후의 값을 참조할 수 있도록 참조할 수 있는 이름을 재 명명할 수 있는 절로 디폴트 값은 OLD와 NEW이다.

-FOR EACH ROW : 행레벨 트리거를 명시하는절

-WHEN (조건) : 테이블/뷰의 행 트리거의 각 행에 대해 제약을 주는 절. 조건은 반드시 괄호로 감싸야 한다.

-TRIGGER BODY : 트리거에 의해 실행될 부분을 정의하는 곳으로, Anonymous PL/SQL 구조가 올 수 있으며, 프로시저를 호출할 수도 있다.(이 경우는 call 문을 사용하며 ;을 붙이지 않는다.)

6. 주요 TRIGGER 유형

1)단순 DML TRIGGER

- BEFORE TRIGGER : 테이블에서 DML 이벤트를 TRIGGER 하기 전
- AFTER TRIGGER : 테이블에서 DML 이벤트를 TRIGGER 한 후
- INSTEAD OF TRIGGER : TRIGGER 문 대신 TRIGGER 본문을 실행하며, 다른 방법으로는 수정이 불가능한 뷰에 사용
- DML TRIGGER는 문장 레벨 트리거와 행 레벨 트리거로 나눈다.

2)STATEMENT LEVEL TRIGGER

-트리거가 설정된 테이블에 트리거 이벤트가 발생하면 여러 행에 대한 변경이 발생하더라도 트리거를 한 번만 발생시키는 방법

-이 방법은 관련된 행의 수에 관계없이 트리거된 문장에 의해 영향을 받게 된다.

-트리거 작업이 영향을 받는 행의 데이터 또는 트리거 이벤트 자체에서 제공하는 데이터에 종속되지 않은 경우에 유용하다.

```
SET SERVEROUTPUT ON
CREATE OR REPLACE TRIGGER trigger_test
BEFORE INSERT OR UPDATE OR DELETE ON emp
BEGIN
   IF TO_CHAR(sysdate, 'DY') IN ('SAT', 'SUN') THEN
      DBMS_OUTPUT.PUT_LINE('주말에는 데이터를 변경할 수 없습니다.');
   END IF;
END;
/

SQL> @demo.sql

Trigger created.


SQL> UPDATE emp SET sal = sal * 1.1;
주말에는 데이터를 변경할 수 없습니다.

14 rows updated.
```

3)**ROW LEVEL TRIGGER**

-트리거가 설정된 테이블에 대해 이벤트가 발생할 때마다 조건을 만족하는 데이터가 복수 개의 행**일** 때 각 행에 대해 트리거**를** 발생시키는 **방**법

-트리거 이벤트의 영향을 받는 행이 없을 경우에는 실행되지 않는다**.**

-영향을 받는 행의 데이터나 트리거 이벤트 자체에서 제공하는 데이터에 트리거 작업이 종속될 경우에 유용하다**.**

-OLD: TRIGGER 가처리한 레코드의 원래 값을 저장

-NEW: 새 값을 포함

```
CREATE TABLE dept_info
(
   deptno  NUMBER(2),
   dname   VARCHAR2(14),
   loc     VARCHAR2(10),
   o_deptno NUMBER(2),
   o_dname  VARCHAR2(14),
   o_loc    VARCHAR2(10),
   gubun    VARCHAR2(10),
   chk_date  DATE
);

CREATE OR REPLACE TRIGGER trigger_test1
AFTER INSERT OR UPDATE OR DELETE ON dept
FOR EACH ROW
BEGIN
   IF INSERTING THEN
      INSERT INTO dept_info(deptno, dname, loc, gubun)
      VALUES (:NEW.deptno, :NEW.dname, :NEW.loc, '입력');
   ELSIF UPDATING THEN
      INSERT INTO dept_info(deptno, dname, loc, o_deptno, o_dname, o_loc
      , gubun)
      VALUES (:NEW.deptno, :NEW.dname, :NEW.loc, :OLD.deptno, :OLD.
      dname, :OLD.loc, '수정');
   ELSIF DELETING THEN
      INSERT INTO dept_info(deptno, dname, loc, gubun)
      VALUES (:OLD.deptno, :OLD.dname, :OLD.loc, '삭제');
   END IF;
END;
/

SQL> @demo.sql

Trigger created.


INSERT INTO dept VALUES (99, 'test1', 'loc1');
INSERT INTO dept VALUES (98, 'test1', 'loc1');
COMMIT;
SELECT * FROM dept_same;

SQL> SELECT * FROM dept_info;

DEPTNO DNAME LOC O_DEPTNO O_DNAME O_LOC GUBUN CHK_DATE
--------------------------------------------------------------------------
99       test1   loc1                                    입력
```

98 test1 loc1 입력

7. **TRIGGER** 변경

1) 트리거의 상태를 비활성화 또는 활성화할 수 있다.

-**TRIGGER**를 생성하면 기본적으로 활성화 상태이다. 하지만, 비즈니스 룰의 변경으로 더이상 트리거를 설정해 둘 필요가 없을 때 상태를 비활성화 상태로 만들 수 있다.

-비활성화 상태에서 다시 활성화 상태로 바꿀 때는 ENABLE 한다.

-**ALTER TRIGGER** trigger_name DISABLE | ENABLE;

2) 해당 테이블과 관련된 모든 트리거의 상태를 비활성화 또는 활성화할 수 있다.

-어떤 테이블에 많은 트리거가 활성화 되어 있는 경우, 개별적으로 바활성화 또는 활성화하기 매우 어려운 경우

-해당 테이블과 관련된 모든 트리거를 활성화 또는 비활성화할 수 있다.

-**ALTER TABLE** trigger_name DISABLE | ENABLE **ALL TRIGGER**;

3) 트리거를 재컴파일 할 수 있다.

-트리거의 내용이 변경되었거나, 트리거가 제대로 실행되지 않을 때 사용자가 직접 재컴파일 시켜서 사용 가능한 상태로 만들 수 있다.

-**ALTER TRIGGER** trigger_name COMPILE;

4) 트리거를 삭제할 수 있다.

-**DROP TRIGGER** trigger_name;

8. **TRIGGER** vs **PROCEDURE**

1) 문법이 다르다.

CREATE PROCEDURE ... vs **CREATE TRIGGER ...**

2) 둘 다 생성하면 소스코드와 실행코드(p-code)가 생성된다.

3) **EXEC** 명령어로 실행 vs 생성 후 자동실행

4) 실행부에서 **COMMIT, ROLLBACK** 실행 가능 vs 사용불가능

9. 사용예

-**UPDATE, INSERT, DELETE** 문에 의해 변경되는 테이블에 대한 트리거가 발생할 때 트리거 내에서 참조되는 테이블이 변경되는 테이블과 같은 경우 다음과 같은 에러가 발생한다.

ORA-04901 **TABLE** DEPT **IS** MUTATING, TRIGGERFUNCTION MAY **NOT** SEE IT

ORA-04088 ERROR DURING EXECUTION **OF TRIGGER**

현재 변경이 진행중인 테이블을 트리거 내에서도 참조하기 때문에 데이터의 무결성을 보장할 수 없게 된다. 사용자의 변경 작업은 더 이상 진행되지 않는다.

```
CREATE OR REPLACE TRIGGER cascade_trigger
BEFORE UPDATE ON emp
FOR EACH ROW
DECLARE
   v_sal emp.sal%TYPE;
BEGIN
   SELECT MAX(sal) INTO v_sal
   FROM emp;
END;
/

SQL> @demo.sql

Trigger created.


UPDATE emp SET sal = sal * 1.1
WHERE empno = 7934;
```

주말에는 데이터를 변경할 수 없습니다.

```
195       UPDATE emp SET sal = sal * 1.1
196           *
197       ERROR at line 1:
198       ORA-04091: table SCOTT.EMP is mutating, trigger/function may not see it
199       ORA-06512: at "SCOTT.CASCADE_TRIGGER", line 7
200       ORA-04088: error during execution of trigger 'SCOTT.CASCADE_TRIGGER'
201
202  10. 사용예2
203       -제약 테이블의 기본키, 외래키, 유일키 칼럼에 대해 데이터를 변경하지 말 것.
204       -부모 테이블의 primary key 와 자식 테이블의 foreign-key가 설정된 2개의 테이블에서
          부모 테이블의 식별키를 변경하면 트리거가 발생하고, 트리거에는 자식 테이블을 변경하는 작업이
          일어난다면 다음과 같은 에러가 발생한다.
205       ORA-04901  TABLE DEPT IS MUTATING, TRIGGERFUNCTION MAY NOT SEE IT
206       즉, 트리거가 설정된 테이블의 식별키, 외래키 칼럼의 데이터는 변경하면 안된다.
207       왜냐하면, 2개의 테이블은 서로 관련된 데이터를 포함하고 있기 때문에 식별키를 변경하면 외래키를
           가진 테이블의 데이터가 무결성을 지킬 수 없기 때문이다.
208
209     CREATE OR REPLACE TRIGGER cascade_trigger1
210     AFTER UPDATE ON dept
211     FOR EACH ROW
212     BEGIN
213       UPDATE emp SET emp.deptno = :NEW.deptno
214       WHERE emp.deptno = :OLD.deptno;
215     END;
216     /
217
218     SQL> @demo.sql
219
220     Trigger created.
221
222     UPDATE dept SET deptno = 30 WHERE deptno = 10;
223     주말에는 데이터를 변경할 수 없습니다.
224     UPDATE dept SET deptno = 30 WHERE deptno = 10
225           *
226     ERROR at line 1:
227     ORA-04091: table SCOTT.EMP is mutating, trigger/function may not see it
228     ORA-06512: at "SCOTT.CASCADE_TRIGGER", line 7
229     ORA-04088: error during execution of trigger 'SCOTT.CASCADE_TRIGGER'
230     ORA-06512: at "SCOTT.CASCADE_TRIGGER1", line 2
231     ORA-04088: error during execution of trigger 'SCOTT.CASCADE_TRIGGER1'
232
233  11. 트리거의 응용 범위
234     1) 보안 : 데이터베이스 내 테이블에 대한 변경을 제한할 수 있다.
235     2) 감사 : 사용자들의 데이터베이스 사용에 대한 모든 내용을 감시할 수 있다.
236     3) 데이터의 무결성 : 테이블에 원치 않는 데이터가 저장되는 것을 방지할 수 있다.
237     4) 테이블의 복제 : 기본 테이블에 대한 똑같은 복사 테이블을 온라인으로 생성, 관리할 수 있다.
238     5) 연속적 작업 수행 : 기본 테이블에 데이터가 입력되면 또 다른 테이블에 데이터를 변경하는
        연속적인 작업을 할 수 있다.
239
240     CREATE TABLE dept_clone
241     AS SELECT * FROM dept;
242
243     CREATE OR REPLACE TRIGGER trigger_test
244     AFTER INSERT OR UPDATE OR DELETE ON dept
245     FOR EACH ROW
246     BEGIN
```

```
247       IF INSERTING THEN
248          INSERT INTO dept_clone(deptno, dname, loc)
249          VALUES (:NEW.deptno, :NEW.dname, :NEW.loc);
250       ELSIF UPDATING THEN
251          UPDATE dept_clone
252          SET dname = :NEW.dname, loc=:NEW.loc
253          WHERE dept_clone.deptno = :OLD.deptno;
254       ELSIF DELETING  THEN
255          DELETE dept_clone
256          WHERE dept_clone.deptno = :OLD.deptno;
257       END IF;
258    END;
259    /
260
261    SQL> @demo.sql
262
263    Trigger created.
264
265    SQL>INSERT INTO dept VALUES (66, '전산과', '서울');
266    SQL>UPDATE dept SET loc = '대전' WHERE deptno = 66;
267
268    SQL>SELECT * FROM dept_clone;
269
270    ----------------------------------------------------------------
271 REM 테이블에 데이터를 입력할 수 있는 시간 지정하기
272    CREATE TABLE t_order
273    (
274       no NUMBER,
275       ord_code  VARCHAR2(10),
276       ord_date DATE
277    );
278
279    --테이블에 데이타를 입력할 때 입력시간이 18:40분에서 18:50분일 경우에만 입력을 허용하고, 그
       외 시간일 경우에는 에러를 발생시키는 trigger 이다.
280    SET SERVEROUTPUT ON
281    CREATE OR REPLACE TRIGGER trigger_test
282    BEFORE INSERT ON t_order
283    BEGIN
284       IF (TO_CHAR(sysdate, 'HH24:MI') NOT BETWEEN '18:40' AND '18:50')
       THEN
285          RAISE_APPLICATION_ERROR(-20100, '허용시간이 아닙니다');
286       END IF;
287    END;
288    /
289
290    SQL> SELECT TO_CHAR(SYSDATE, 'HH24:MI') AS NOW
291         2  FROM dual;
292
293         NOW
294         ----------
295         09:10
296
297    SQL> INSERT INTO t_order
298          2  values(2, 'C200', SYSDATE);
299            INSERT INTO t_order
300               *
```

```
ERROR at line 1:
ORA-20100: 허용시간이 아닙니다
ORA-06512: at "SCOTT.TRIGGER_TEST", line 3
ORA-04088: error during execution of trigger 'SCOTT.TRIGGER_TEST'


REM 테이블에 입력될 데이터 값을 지정하고 그 값 이외에는 에러를 발생시키는 TRIGGER 를 생성하라.
CREATE TABLE t_order
(
   no NUMBER,
   ord_code  VARCHAR2(10),
   ord_date DATE
);

SET SERVEROUTPUT ON
CREATE OR REPLACE TRIGGER trigger_test1
BEFORE INSERT ON t_order
FOR EACH ROW --행레벨 트리거
BEGIN
   IF (:NEW.ord_code) NOT IN ('C100') THEN
      RAISE_APPLICATION_ERROR(-20200, '제품 코드가 틀립니다.');
   END IF;
END;
/

SQL> @demo.sql

Trigger created.

SQL> insert into t_order
  2  values (2, 'C100',SYSDATE);
insert into t_order
            *
ERROR at line 1:
ORA-20100: 허용시간이 아닙니다
ORA-06512: at "SCOTT.TRIGGER_TEST", line 3
ORA-04088: error during execution of trigger 'SCOTT.TRIGGER_TEST'


SQL> DROP TRIGGER trigger_test;

Trigger dropped.

SQL> INSERT INTO t_order
  2  VALUES (2, 'C100', SYSDATE);

1 row created.

SQL> INSERT INTO t_order
  2  VALUES (3, 'C200', SYSDATE);
insert into t_order
            *
ERROR at line 1:
ORA-20200: 제품 코드가 틀립니다.
ORA-06512: at "SCOTT.TRIGGER_TEST1", line 3
ORA-04088: error during execution of trigger 'SCOTT.TRIGGER_TEST1'
```

```
REM TRIGGER 의 작동조건을 WHEN 절로 더 자세히 정의
   --'C500'인 제품에 대해서만 19:30분부터 19:35분까지만 입력을 허용하는 트리거
   --그외 다른 제품은 시간에 상관없이 정상적으로 입력

   SET SERVEROUTPUT ON
   CREATE OR REPLACE TRIGGER trigger_test2
   BEFORE INSERT ON t_order
   FOR EACH ROW --행레벨 트리거
   WHEN (NEW.ord_code = 'C500')
   BEGIN
      IF (TO_CHAR(SYSDATE, 'HH24:MI') NOT BETWEEN '19;30' AND '19:35')
      THEN
         RAISE_APPLICATION_ERROR(-20300, 'C500 제품의 입력 허용 시간이 아닙니다.');
      END IF;
   END;
   /

   SQL> SELECT TO_CHAR(SYSDATE, 'HH24:MI') NOW FROM dual;

   NOW
   ----------
   14:50

   SQL> INSERT INTO t_order
     2  VALUES (1, 'C500', SYSDATE);
   INSERT INTO t_order
               *
   ERROR at line 1:
   ORA-20300: C500 제품의 입력 허용 시간이 아닙니다.
   ORA-06512: at "SCOTT.TRIGGER_TEST2", line 3
   ORA-04088: error during execution of trigger 'SCOTT.TRIGGER_TEST2'


   SQL> INSERT INTO t_order
     2  VALUES (2, 'C600', SYSDATE);
   INSERT INTO t_order
               *
   ERROR at line 1:
   ORA-20200: 제품 코드가 틀립니다.
   ORA-06512: at "SCOTT.TRIGGER_TEST1", line 3
   ORA-04088: error during execution of trigger 'SCOTT.TRIGGER_TEST1'


   SQL> INSERT INTO t_order
     2  VALUES (2, 'C100', SYSDATE);

   1 row created.
```