SET SERVEROUTPUT ON

--+ TRIGGER

-- : INSERT, UPDATE, DELETE 문이 TABLE에 대해 행해질 때 묵시적으로 수행되는 PROCEDURE

-- : 테이블과는 별도의 객체 (별도로 DATABASE에 저장)

-- : VIEW에 대해서가 아니라 TABLE에 관해서만 정의할 수 있다.

-- : 이벤트가 실행되면 자동으로 실행되는 것 (암시적 수행)

--

-- - 행 트리거 : 컬럼의 각 행의 데이터 행 변화가 생길 때마다 실행되며, 그 데이터 행의 실제값을 제어할 수 있음

--

-- - 문장 트리거 : 트리거 이벤트에 대해 단 한번 실행, 컬럼의 각 데이터행을 제어할 수 없음

--

-- [형식]

-- CREATE OR REPLACE TRIGGER [TRIGGER NAME]

-- BEFORE|AFTER (BEFORE : DML문 실행되기 전, AFTER : DML문 실행 된 후)

-- [TRIGGER EVENT] ON [TABLE NAME]

--

-- [FOR EACH ROW] {행 트리거}

-- [WHEN CONDITION]

CREATE TABLE COPY\_DEPT

AS

SELECT \* FROM DEPT;

CREATE OR REPLACE TRIGGER TRIGGER\_EX

BEFORE

UPDATE ON COPY\_DEPT

FOR EACH ROW

BEGIN

DBMS\_OUTPUT.PUT\_LINE('변경 전 부서이름 값: '|| :OLD.DNAME);

DBMS\_OUTPUT.PUT\_LINE('변경 후 부서이름 값: '|| :NEW.DNAME);

END;

/

UPDATE COPY\_DEPT

SET DNAME = '영업1'

WHERE DEPTNO = 10;

변경 전 부서이름 값: 영업1

변경 후 부서이름 값: 영업1

1 행 이(가) 업데이트되었습니다.

CREATE OR REPLACE TRIGGER AVG\_TRIGGER

BEFORE -- 이벤트가 일어날 때 INSERT 값이 들어오기 전의 정보로 실행해라

INSERT OR UPDATE ON CEMP

FOR EACH ROW

DECLARE

AVG\_SAL NUMBER;

BEGIN

SELECT ROUND(AVG(SAL), 2) INTO AVG\_SAL

FROM CEMP;

DBMS\_OUTPUT.PUT\_LINE('사원 급여 평균: '||AVG\_SAL);

END;

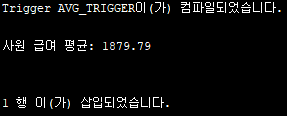
/

INSERT INTO CEMP (EMPNO, ENAME, SAL)

VALUES (8888, 'HONG', 10000); --평균 1879.79 (전)

SELECT ROUND(AVG(SAL),2)

FROM CEMP; --평균 2421.14 (후)





--DEPT 테이블에 값을 넣으면 CDEPT 테이블에도 값이 들어가게 하고 싶음

CREATE TABLE CDEPT

AS

SELECT \* FROM DEPT;

CREATE OR REPLACE TRIGGER INS\_DEPT

AFTER

INSERT ON DEPT

FOR EACH ROW

-- 변수 선언

-- 1. 원테이블에서 변화가 있는 부분을 가져오기

-- 2. 변수 할당: 변화가 있는 컬럼 수 만큼 할당(동적으로 선언)

-- 3. 다른 테이블에 값을 넣기

-- : TRIGGER가 알아서 자동으로 수행해 줌

BEGIN

-- DB에서 원테이블에 최근에 입력한 값을 가져오기 (TRIGGER가 자동으로 인식)

-- 개발자는 최종 목적을 위한 로직만 처리하면 됨

INSERT INTO CDEPT

VALUES (:NEW.DEPTNO, :NEW.DNAME, :NEW.LOC);

END;

/

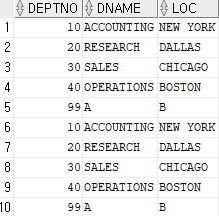
INSERT INTO DEPT

VALUES (99,'A','B'); --INSERT 후 COMMIT하기

SELECT \* FROM DEPT

UNION ALL

SELECT \* FROM CDEPT;



CREATE OR REPLACE TRIGGER UPDATE\_DEPT

AFTER UPDATE ON DEPT

FOR EACH ROW

BEGIN

UPDATE CDEPT

SET DNAME = :NEW.DNAME, LOC = :NEW.LOC

WHERE DEPTNO = :OLD.DEPTNO;

END;

/

UPDATE DEPT

SET DNAME = 'C', LOC='D'

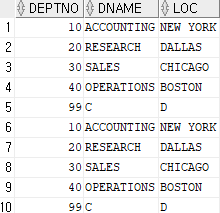
WHERE DEPTNO = 99;

SELECT \* FROM DEPT

UNION ALL

SELECT \* FROM CDEPT;

(DEPT 업데이트 했더니 CDEPT도 업데이트 됨)



--실습 : DEPT테이블에 부서번호 99번을 삭제하는 기능 수행 시 CDEPT 테이블의 99번 부서도 삭제되도록 해보세요

--DEL\_DEPT

CREATE OR REPLACE TRIGGER DEL\_DEPT

AFTER DELETE ON DEPT

FOR EACH ROW

BEGIN

DELETE CDEPT

WHERE DEPTNO = :OLD.DEPTNO;

END;

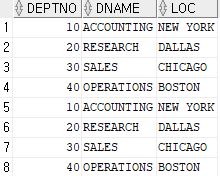
/

DELETE DEPT WHERE DEPTNO=99;

SELECT \* FROM DEPT

UNION ALL

SELECT \* FROM CDEPT;



--통합 버전 트리거

CREATE OR REPLACE TRIGGER DML\_TRI

AFTER INSERT OR UPDATE OR DELETE ON DEPT

FOR EACH ROW

BEGIN

IF INSERTING THEN

INSERT INTO CDEPT

VALUES (:NEW.DEPTNO, :NEW.DNAME, :NEW.LOC);

ELSIF UPDATING THEN

UPDATE CDEPT

SET DNAME = :NEW.DNAME, LOC = :NEW.LOC

WHERE DEPTNO = :OLD.DEPTNO;

ELSIF DELETING THEN

DELETE CDEPT

WHERE DEPTNO = :OLD.DEPTNO;

END IF;

END;

/

위의 트리거들을 합친 것, 근데 CDEPT에는 값이 2번씩 들어가는 문제가 있음