

PÉLDÁK II.

/GYAKORLATI PÉLDÁK EMP/DEPT TÁBLÁKON/

1. példa (System Events)

(Kliens események: LOGON, LOGOFF, BEFORE CREATE, AFTER CREATE, BEFORE ALTER, AFTER ALTER, BEFORE DROP, AFTER DROP)

Hozzuk létre a segédtáblát!

```
CREATE TABLE LOGON(  
    NEV VARCHAR(20) ,  
    DATUM DATE  
);
```

Hozzuk létre a triggereket! (felhasználónév helyére h-s azonosító)

```
CREATE OR REPLACE TRIGGER On_Logon  
    AFTER LOGON  
    ON felhasználónév.Schema  
BEGIN  
    INSERT INTO LOGON VALUES (USER || ' belepett', SYSDATE);  
END;
```

```
CREATE OR REPLACE TRIGGER On_Logoff  
    BEFORE LOGOFF  
    ON felhasználónév.Schema  
BEGIN  
    INSERT INTO LOGON VALUES (USER || ' kilepett', SYSDATE);  
END;
```

Kérjük le a logon tábla tartalmát (a dátumot konvertáljuk át)!

```
select nev, to_char(datum, 'DD/MM/YYYY HH24:MI:SS') as "DATUM"  
from logon;
```

2. példa

Hozzunk létre egy olyan utasításszintű BEFORE triggert, amely megakadályozza a munkaidőn kívüli adatmanipulációkat az emp táblán!

```
CREATE OR REPLACE TRIGGER NeNyuljHozza  
    BEFORE DELETE OR INSERT OR UPDATE ON emp  
BEGIN  
    IF TO_CHAR(SYSDATE, 'HH24:MI') > '14:00'  
    THEN  
        RAISE_APPLICATION_ERROR(-20111, 'Csak munkaidoben szabad  
az adatbazis!');  
    END IF;  
END;
```

Próbáljuk meg beszúrni egy sort!

```
insert into emp values (1234, 'kiss', 'clerk', 1111, '99-MAJ-20', 1200, NULL, 10)
```

3. példa

Oldjuk meg az előző feladatot oly módon, hogy azt is kiiradjuk, milyen műveletet kíséreltek meg végrehajtani munkaidőn kívül.

```
CREATE OR REPLACE TRIGGER NeNyuljHozza
BEFORE DELETE OR INSERT OR UPDATE ON emp
BEGIN
    IF TO_CHAR(SYSDATE, 'HH24:MI') > '14:00'
    THEN
        IF DELETING THEN
            RAISE_APPLICATION_ERROR(-20211, 'Csak munkaidoben szabad
adatot torolni!');
        ELSIF INSERTING THEN
            RAISE_APPLICATION_ERROR(-20212, 'Csak munkaidoben szabad
adatot bevinni!');
        ELSE
            RAISE_APPLICATION_ERROR(-20213, 'Csak munkaidoben szabad
adatot modositani!');
        END IF;
    END IF;
END;
```

Próbáljunk meg törölni egy sort!

```
DELETE FROM emp WHERE UPPER(ename) = UPPER('Smith')
```

4. példa

Írjunk trigger, mely megakadályozza az elnökre (president) vonatkozó törlő, beszúró és adatszűrő utasítások működését!

```
CREATE OR REPLACE TRIGGER KiveveAzElnok
BEFORE DELETE OR INSERT OR UPDATE ON emp
FOR EACH ROW
WHEN (UPPER(OLD.job) = 'PRESIDENT' OR
      UPPER(NEW.job) = 'PRESIDENT')
BEGIN
    IF DELETING THEN
        RAISE_APPLICATION_ERROR(-20001, 'Az elnokot nem lehet
torolni!');
    ELSIF INSERTING THEN
        RAISE_APPLICATION_ERROR(-20002, 'Elnokot nem lehet
beszurni!');
    ELSIF UPDATING THEN
        RAISE_APPLICATION_ERROR(-20003, 'Elnok adatok nem
modosithatok!');
    END IF;
END;
```

Próbáljunk meg frissíteni elnököt!

```
UPDATE emp SET sal = sal - 555 WHERE deptno = 10
```

Próbáljunk meg beszúrni elnököt!

```
insert into emp values (5678, 'kiss', 'president', NULL,  
SYSDATE, 6000, NULL, 10)
```

Próbáljunk meg törölni elnököt!

```
delete from emp where deptno = 10
```