|  |
| --- |
| CREATE OR REPLACE TRIGGER TRIGGER1  BEFORE INSERT OR UPDATE OF SOZIO\_IZENA ON SOZIOAK  declare  a\_kop INTEGER := 0;  a\_id\_sozioa sozioa.id\_sozioa%type;  a\_id\_asoziaioa asoziazioa.id\_asoziazioa%type;  BEGIN  if inserting then    if (:new.kolmena\_kantitatea >600 and :new.kolmena\_kantitatea< 800 ) or (:new.erle\_kantitatea <600000 and :new.erle\_kantitatea >400000) then  if (:new.kolmena\_kantitatea >600 and :new.kolmena\_kantitatea< 800 )= false and (:new.erle\_kantitatea <600000 and :new.erle\_kantitatea >400000)then  :new.kolmena\_kantitatea:= 600;    end if;  else    raise\_application\_error(-20004, 'Errorea: Kolmena eta erle kantitateak ez dituzte baldintzak betetzen.');    end if;        elsif updating then    select id\_sozioak into a\_id\_sozioa from sozioak where :new.izena =izena;  select count(\*) into a\_kop from ze\_ezti where id\_sozioa = a\_id\_sozioa;    if a\_kop > 5 then  :new.sozio\_izena:= :new.sozio\_izena +'\_TOP';  select id\_asoziazioa into a\_id\_asoziazioa from asoziazio\_parte where id\_sozioa=a\_id\_sozioa;  update asoziazioak set asoziazio\_izena = asoziazio\_izena+'\_TOP' where id\_asoziazioa=a\_id\_asoziaioa;    end if;    end if;  NULL;  END; |

Lehenengo, kontuan artu behar dena, bi “if” orokor daudela da. Triggerrean “insert” eta “update” ezberdindu behar dira eta. Ondorioz bakoitzaren arabera balio ezberdinak konparatu eta aldatuko dira.

|  |
| --- |
| CREATE OR REPLACE TRIGGER TRIGGER1  BEFORE INSERT OR UPDATE OF SOZIO\_IZENA ON SOZIOAK  declare  BEGIN  if inserting then  elsif updating then    end if;  NULL;  END; |

Lehenengo insert-a osatuko da. Ariketan eskatzen duena kontuan hartuta bi if jarri baehar da, bata bestearen barruan. Bertan bi baldintza bete behar dira: kolmena kantitatea 600 eta 800 artean egotea edo erle kantitatea 400000 eta 600000 artean egotea. Horretaz gain “betwen” ez erabiltzea eskatzen da orduan “and” eta “or” erabili behar da.

|  |
| --- |
| CREATE OR REPLACE TRIGGER TRIGGER1  BEFORE INSERT OR UPDATE OF SOZIO\_IZENA ON SOZIOAK  declare  BEGIN  if inserting then    if (:new.kolmena\_kantitatea >600 and :new.kolmena\_kantitatea< 800 ) or (:new.erle\_kantitatea <600000 and :new.erle\_kantitatea >400000) then  end if;    elsif updating then    end if;  NULL;  END; |

Ariketak esaten duenez, kolmena kantitatea ez bada betetzen baina erle kantitatea bai, kolmena kantitatea 600 jarriko da. Horretarako hasierako baldintza beteko balitz beste baldintza bat egon behar da horren barruan. Honek kolmena kantitatea gezurra dela eta erle kantitatea betetzen dela kudeatzeko.

|  |
| --- |
| CREATE OR REPLACE TRIGGER TRIGGER1  BEFORE INSERT OR UPDATE OF SOZIO\_IZENA ON SOZIOAK  declare  a\_kop INTEGER := 0;  a\_id\_sozioa sozioa.id\_sozioa%type;  a\_id\_asoziaioa asoziazioa.id\_asoziazioa%type;  BEGIN  if inserting then    if (:new.kolmena\_kantitatea >600 and :new.kolmena\_kantitatea< 800 ) or (:new.erle\_kantitatea <600000 and :new.erle\_kantitatea >400000) then  if (:new.kolmena\_kantitatea >600 and :new.kolmena\_kantitatea< 800 )= false and (:new.erle\_kantitatea <600000 and :new.erle\_kantitatea >400000)then  :new.kolmena\_kantitatea:= 600;    end if;      end if;        elsif updating then      end if;  NULL;  END; |

Baldintza hau ez bada betetzen errore bat agertuko da, horretarako “rise\_aplication\_error” erabili da.

|  |
| --- |
| CREATE OR REPLACE TRIGGER TRIGGER1  BEFORE INSERT OR UPDATE OF SOZIO\_IZENA ON SOZIOAK  declare  a\_kop INTEGER := 0;  a\_id\_sozioa sozioa.id\_sozioa%type;  a\_id\_asoziaioa asoziazioa.id\_asoziazioa%type;  BEGIN  if inserting then    if (:new.kolmena\_kantitatea >600 and :new.kolmena\_kantitatea< 800 ) or (:new.erle\_kantitatea <600000 and :new.erle\_kantitatea >400000) then  if (:new.kolmena\_kantitatea >600 and :new.kolmena\_kantitatea< 800 )= false and (:new.erle\_kantitatea <600000 and :new.erle\_kantitatea >400000)then  :new.kolmena\_kantitatea:= 600;    end if;  else    raise\_application\_error(-20004, 'Errorea: Kolmena eta erle kantitateak ez dituzte baldintzak betetzen.');    end if;        elsif updating then        end if;  NULL;  END; |

Lehenengo “if” orokorrarekin bukatuta Update-ko baldintza egitea falta da. Ariketaren arabera bost ezti mota baino gehiago saltzen duten sozioen izenean “\_TOP” jarri behar da. Horretarako sozio bakoitzak saltzen dituen ezti motak kontatu behar da. Horretarako ezti produktuen mota erabiliko da. Horretarako aldatutako izenaren id\_sozioa lortu da eta horrekin ezti motak kontatu “count()” erabiliz. Honek biak aldagai baten gordeko dira; a\_id\_sozioa ezti motak jakitzeko erabiliko da bertan sartutako izenaren id-a gordeko da eta a\_kop “count” erabiliz ezti mota kopurua gordeko da.

|  |
| --- |
| CREATE OR REPLACE TRIGGER TRIGGER1  BEFORE INSERT OR UPDATE OF SOZIO\_IZENA ON SOZIOAK  declare  a\_kop INTEGER := 0;  a\_id\_sozioa sozioa.id\_sozioa%type;  a\_id\_asoziaioa asoziazioa.id\_asoziazioa%type;  BEGIN  if inserting then    if (:new.kolmena\_kantitatea >600 and :new.kolmena\_kantitatea< 800 ) or (:new.erle\_kantitatea <600000 and :new.erle\_kantitatea >400000) then  if (:new.kolmena\_kantitatea >600 and :new.kolmena\_kantitatea< 800 )= false and (:new.erle\_kantitatea <600000 and :new.erle\_kantitatea >400000)then  :new.kolmena\_kantitatea:= 600;    end if;  else    raise\_application\_error(-20004, 'Errorea: Kolmena eta erle kantitateak ez dituzte baldintzak betetzen.');    end if;        elsif updating then    select id\_sozioak into a\_id\_sozioa from sozioak where :new.izena =izena;  select count(\*) into a\_kop from ze\_ezti where id\_sozioa = a\_id\_sozioa;      end if;  NULL;  END; |

Aldagaietan balioa edukita baldintza bat sortu mota kopurua bost baino handiagoa den jakiteko eta horrela izenari “\_TOP” gehitzeko.

|  |
| --- |
| CREATE OR REPLACE TRIGGER TRIGGER1  BEFORE INSERT OR UPDATE OF SOZIO\_IZENA ON SOZIOAK  declare  a\_kop INTEGER := 0;  a\_id\_sozioa sozioa.id\_sozioa%type;  a\_id\_asoziaioa asoziazioa.id\_asoziazioa%type;  BEGIN  if inserting then    if (:new.kolmena\_kantitatea >600 and :new.kolmena\_kantitatea< 800 ) or (:new.erle\_kantitatea <600000 and :new.erle\_kantitatea >400000) then  if (:new.kolmena\_kantitatea >600 and :new.kolmena\_kantitatea< 800 )= false and (:new.erle\_kantitatea <600000 and :new.erle\_kantitatea >400000)then  :new.kolmena\_kantitatea:= 600;    end if;  else    raise\_application\_error(-20004, 'Errorea: Kolmena eta erle kantitateak ez dituzte baldintzak betetzen.');    end if;        elsif updating then    select id\_sozioak into a\_id\_sozioa from sozioak where :new.izena =izena;  select count(\*) into a\_kop from ze\_ezti where id\_sozioa = a\_id\_sozioa;    if a\_kop > 5 then  :new.sozio\_izena:= :new.sozio\_izena +'\_TOP';      end if;    end if;  NULL;  END; |

Baina ariketan esaten duenez, sozioak parte hartzen duen asoziazio guztien izenean ere “\_TOP” jarri behar da. Horretarako gordetako id-a erabiliko da id\_asoziazio jakiteko. Ondoren hori gorde eta izena atera ahalko da eta hau eraldatu.

|  |
| --- |
| CREATE OR REPLACE TRIGGER TRIGGER1  BEFORE INSERT OR UPDATE OF SOZIO\_IZENA ON SOZIOAK  declare  a\_kop INTEGER := 0;  a\_id\_sozioa sozioa.id\_sozioa%type;  a\_id\_asoziaioa asoziazioa.id\_asoziazioa%type;  BEGIN  if inserting then    if (:new.kolmena\_kantitatea >600 and :new.kolmena\_kantitatea< 800 ) or (:new.erle\_kantitatea <600000 and :new.erle\_kantitatea >400000) then  if (:new.kolmena\_kantitatea >600 and :new.kolmena\_kantitatea< 800 )= false and (:new.erle\_kantitatea <600000 and :new.erle\_kantitatea >400000)then  :new.kolmena\_kantitatea:= 600;    end if;  else    raise\_application\_error(-20004, 'Errorea: Kolmena eta erle kantitateak ez dituzte baldintzak betetzen.');    end if;        elsif updating then    select id\_sozioak into a\_id\_sozioa from sozioak where :new.izena =izena;  select count(\*) into a\_kop from ze\_ezti where id\_sozioa = a\_id\_sozioa;    if a\_kop > 5 then  :new.sozio\_izena:= :new.sozio\_izena +'\_TOP';  select id\_asoziazioa into a\_id\_asoziazioa from asoziazio\_parte where id\_sozioa=a\_id\_sozioa;  update asoziazioak set asoziazio\_izena = asoziazio\_izena+'\_TOP' where id\_asoziazioa=a\_id\_asoziaioa;    end if;    end if;  NULL;  END; |

Azkenik triggerra log trigger bat bihurtu behar da. Horretarako beste bi tabla berri sortu behar dira bakoitzeko aldaketak gordetzeko: Sozioak\_Aldaketak eta Asoziazioa\_Aldaketak dira. Honek bi zutabe gehiago edukiko dute HasieraData eta BukaeraData deiturikoak. Bertan aldaketa bakoitzeko balioak eta honen orduak gordeko dira.

|  |
| --- |
| CREATE TABLE sozioak\_aldaketak(  id\_sozioa NUMBER(10) ,  id\_zuzendaria NUMBER(10),  erle\_kantitatea NUMBER(10) DEFAULT 0,  kolmena\_kantitatea NUMBER(10) DEFAULT 0,  sozio\_izena VARCHAR2(25) NOT NULL,  sozio\_abizena VARCHAR2(25) NOT NULL,  NAN VARCHAR2(25),  telefonoa NUMBER(15),  jaiote\_eguna DATE,  email VARCHAR2(50),  CONSTRAINT chk\_NAN\_length CHECK (LENGTH(NAN) BETWEEN 9 AND 10)  );  CREATE TABLE asoziazioak\_aldaketak(  id\_asoziazioa NUMBER(10),  asoziazio\_izena VARCHAR2(40) NOT NULL ,  herrialdea VARCHAR2(25) NOT NULL,  HasieraData TIMESTAMP,  BukaeraData TIMESTAMP  ); |