Taldea 5 2024/3/15 Aritz Lekube, IñigoArrizabalo eta Julen Garcia Mata

2.4 Mugarria

[Triggerra 3](#_Toc163649970)

[Triggera pausus pausu 4](#_Toc163649971)

[Log Triggera sortu 9](#_Toc163649972)

[Irudia 1-Erlauntz eta erle kopurau betetzen ez duen sozioa txertatu. 12](#_Toc163716889)

[Irudia 2- Erlauntz eta erle kantitate egokiak dituen sozioa txertatu. 13](#_Toc163716890)

[Irudia 3-Txertatutako sozioa sozioak taulan. 13](#_Toc163716891)

[Irudia 4- Sozioa historikoa txertatutako sozioa, log triggerraren bidez. 13](#_Toc163716892)

[Irudia 5- Erlauntz kantitatea betetzen ez duen sozioa txertatu. 14](#_Toc163716893)

[Irudia 6-Sozioa sozioak taulan txertatu eta triggeraren ondorioz eurlantz koprua ezarri. 14](#_Toc163716894)

[Irudia 7-Sozioak\_historikoa taulan erregistrua log triggerraren ondorioz gehitu. 14](#_Toc163716895)

# Triggerra

|  |
| --- |
| create or replace NONEDITIONABLE TRIGGER Trigger\_Mugarria  BEFORE INSERT OR UPDATE OF SOZIO\_IZENA ON SOZIOAK  For each row  declare  a\_kop INTEGER := 0;  a\_id\_sozioa NUMBER;  a\_id\_asoziazioa NUMBER;  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 count(\*) into a\_kop from ze\_ezti where id\_sozioa = :old.id\_sozioa;    if a\_kop >= 5 then  :new.sozio\_izena:= :new.sozio\_izena || '\_TOP';    update asoziazioak set asoziazio\_izena = asoziazio\_izena || '\_TOP' where id\_asoziazioa in (select id\_asoziazioa from asoziazio\_parte where asoziazioak.id\_asoziazioa=asoziazio\_parte.id\_asoziazioa);    end if;    end if;  NULL;  END; |

## Triggera pausus pausu

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 baino gehiago saltzen duten sozioen izenean “\_TOP” jarri behar da. Hau ze\_ezti taulatik ikusiko da, bertan izena aldatutako sozioaren id konparatuko da, “count” erabiliz zenbat aldiz kontatu into batekin a\_kop en gorde.

|  |
| --- |
| 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 count(\*) into a\_kop from ze\_ezti where id\_sozioa = :old.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 count(\*) into a\_kop from ze\_ezti where id\_sozioa = :old.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 update bat erabiliko da eta update baten inner join ezin denez egin, azpikontsultak erabili dira.

|  |
| --- |
| 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.sozioa\_izena =sozio\_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';  update asoziazioak set asoziazio\_izena = asoziazio\_izena || '\_TOP' where id\_asoziazioa in (select id\_asoziazioa from asoziazio\_parte where asoziazioak.id\_asoziazioa=asoziazio\_parte.id\_asoziazioa);    end if;    end if;  NULL;  END; |

# 

# Log Triggera sortu

Azkenik triggerra log trigger bat bihurtu behar da. Horretarako beste bi taula 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 data eta orduak gordeko dira.

|  |
| --- |
| CREATE TABLE sozioak\_historikoa(  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\_1 CHECK (LENGTH(NAN) BETWEEN 9 AND 10)  );  CREATE TABLE asoziazioak\_historikoa(  id\_asoziazioa NUMBER(10),  asoziazio\_izena VARCHAR2(40) NOT NULL ,  herrialdea VARCHAR2(25) NOT NULL,  HasieraData TIMESTAMP,  BukaeraData TIMESTAMP  ); |

Azkenik loh trigger-a osatu da. Hau apunteetan bidalitakoa moldatuta egin da. Hurrengo log trigerra da moldatu behar dena.

|  |
| --- |
| **CREATE** **OR** **REPLACE** **TRIGGER** TableTrigger  **AFTER** **INSERT** **OR** **UPDATE** **OR** **DELETE** **ON** OriginalTable  **FOR** **EACH** **ROW**  **DECLARE** Now **TIMESTAMP**;  **BEGIN**  **SELECT** **CURRENT\_TIMESTAMP** **INTO** Now **FROM** Dual;  **UPDATE** HistoryTable  **SET** EndDate = Now  **WHERE** EndDate **IS** **NULL**  **AND** Column1 = :**OLD**.Column1;  **IF** :**NEW**.Column1 **IS** **NOT** **NULL** **THEN**  **INSERT** **INTO** HistoryTable (Column1, Column2, ..., Columnn, StartDate, EndDate)  **VALUES** (:**NEW**.Column1, :**NEW**.Column2, ..., :**NEW**.Columnn, Now, **NULL**);  **END** **IF**;  **END**; |

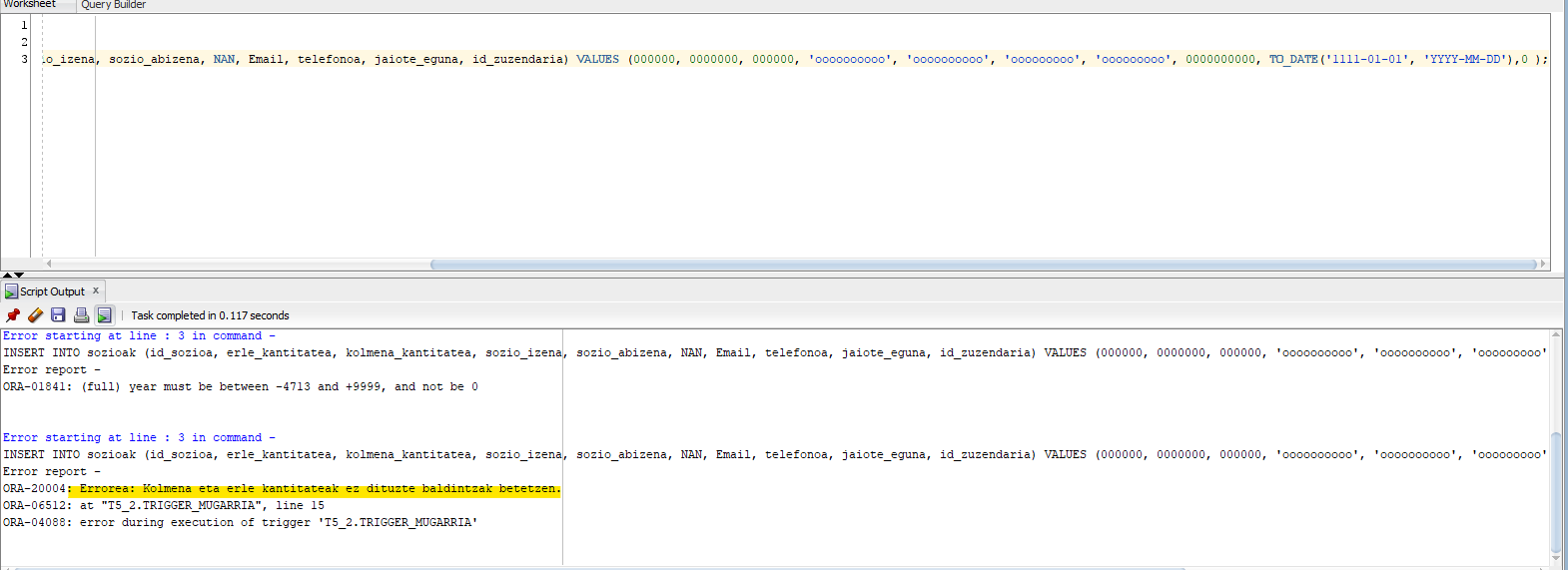
Lehenengo taularen izena aldatu, hemen bi taula kontuan izan behar dugu: aldaketa egiten den tabua eta aldaketa horren balio zaharrak eta berriak gordetzeko taula. Lehena sozioak taula izango da eta bergan gertatzen diren aldaketan sozioak\_historikoa taulan agertuko dira bertan aldaketen datak jarriz.

|  |
| --- |
| create or replace NONEDITIONABLE TRIGGER LOGTRIGGER\_SOZIOAK  AFTER INSERT OR UPDATE OR DELETE ON SOZIOAK  FOR EACH ROW  DECLARE Now TIMESTAMP;  BEGIN  SELECT CURRENT\_TIMESTAMP INTO Now FROM Dual;  UPDATE sozioak\_historikoa  SET BukaeraData = Now  WHERE BukaeraData IS NULL  AND sozio\_izena = :OLD.sozio\_izena;  IF :NEW.sozio\_izena IS NOT NULL THEN  INSERT INTO sozioak\_historikoa  VALUES (:NEW.id\_sozioa, :NEW.id\_zuzendaria,:NEW.erle\_kantitatea, :NEW.kolmena\_kantitatea, :NEW.sozio\_izena,:NEW.sozio\_abizena, :NEW.NAN, :NEW.telefonoa, :NEW.jaiote\_eguna, :NEW.email, Now, NULL);  END IF;  END; |

Aldaketan bi tauletan egiten direnez bi log trigger sortzea behar da. Haurrekoak sozioak taulan egiten diren aldaketak kudeatzeaz arduratzen da. Baina asoziazioak taula ere aldatu egiten da beraz beste log trigger bat sortu da hori kudeatzeko. Hau egiteko aurreko log trigger-a kontuan hartu da eta honen izenak eta zutabeak moldatu.

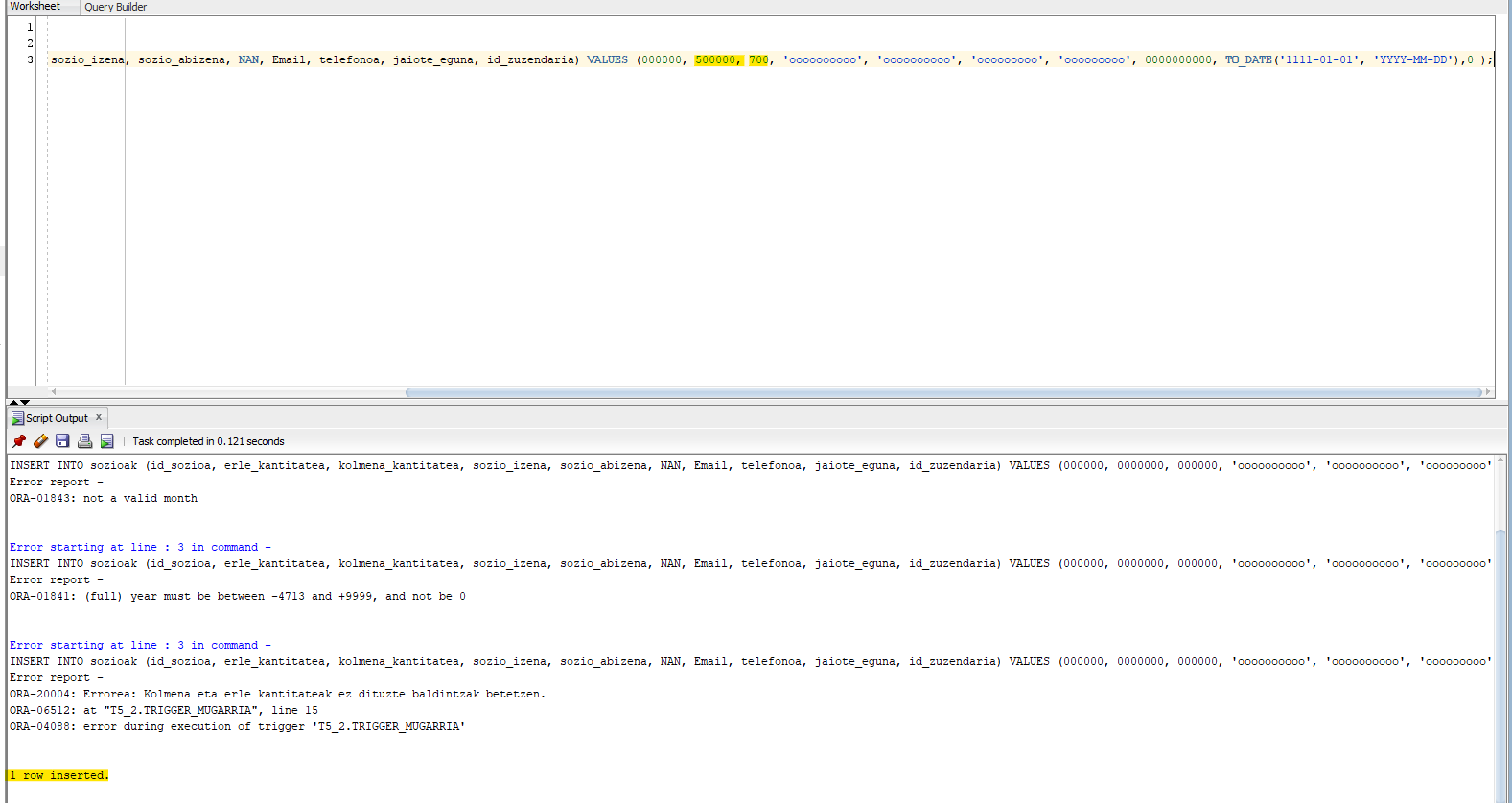
|  |
| --- |
| create or replace NONEDITIONABLE TRIGGER LOGTRIGGER\_ASOZIAZIOAK  AFTER INSERT OR UPDATE OR DELETE ON asoziazioak  FOR EACH ROW  DECLARE Now TIMESTAMP;  BEGIN  SELECT CURRENT\_TIMESTAMP INTO Now FROM Dual;  UPDATE asoziazioak\_historikoa  SET BukaeraData = Now  WHERE BukaeraData IS NULL  AND asoziazio\_izena = :OLD.asoziazio\_izena;  IF :NEW.asoziazio\_izena IS NOT NULL THEN  INSERT INTO asoziazioak\_historikoa  VALUES (:NEW.id\_asoziazioa, :NEW.asoziazio\_izena,:NEW.herrialdea, Now, NULL);  END IF;  END; |

Insert bat egiten bada eta erlauntz kantitateta eta erle kantitatea ez badira betetzen, errorea agertzen da.

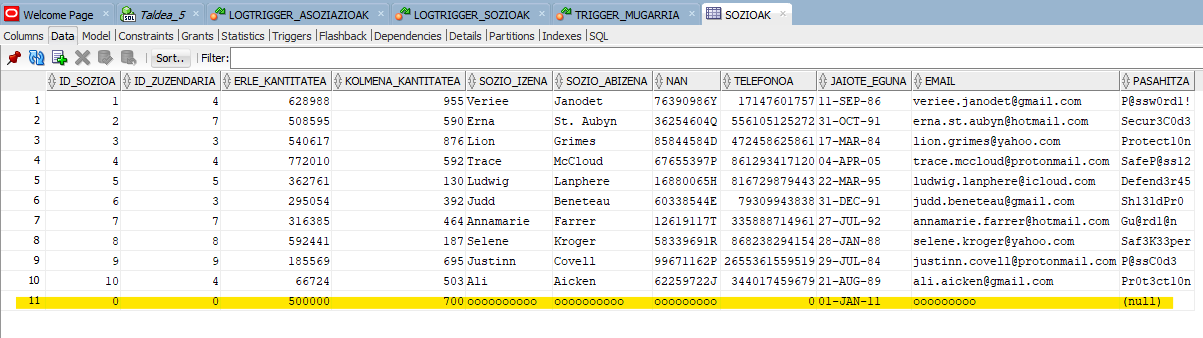


Irudia 1-Erlauntz eta erle kopurau betetzen ez duen sozioa txertatu.

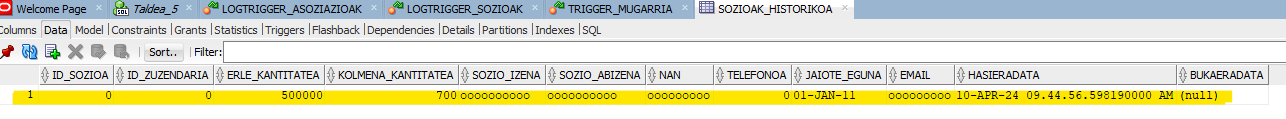
Kolmena kantitatea eta erle kantitatea lehenengo baldintza betetzen badu baina bigarren ez hau arazorik gabe txertatuko da, eta sozioak historikoan ere ongi txertatuko da.



Irudia 2- Erlauntz eta erle kantitate egokiak dituen sozioa txertatu.

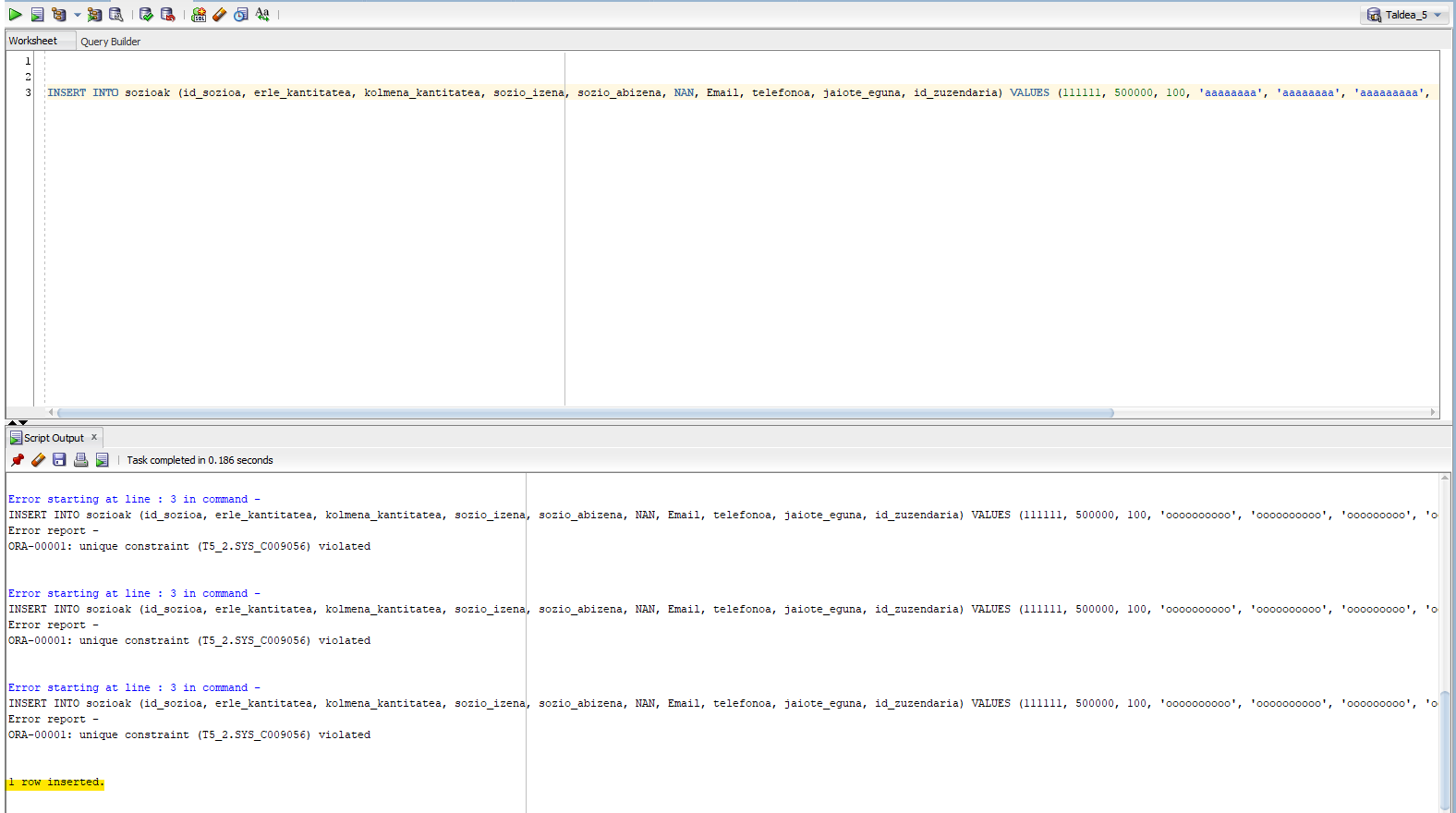


Irudia 3-Txertatutako sozioa sozioak taulan.



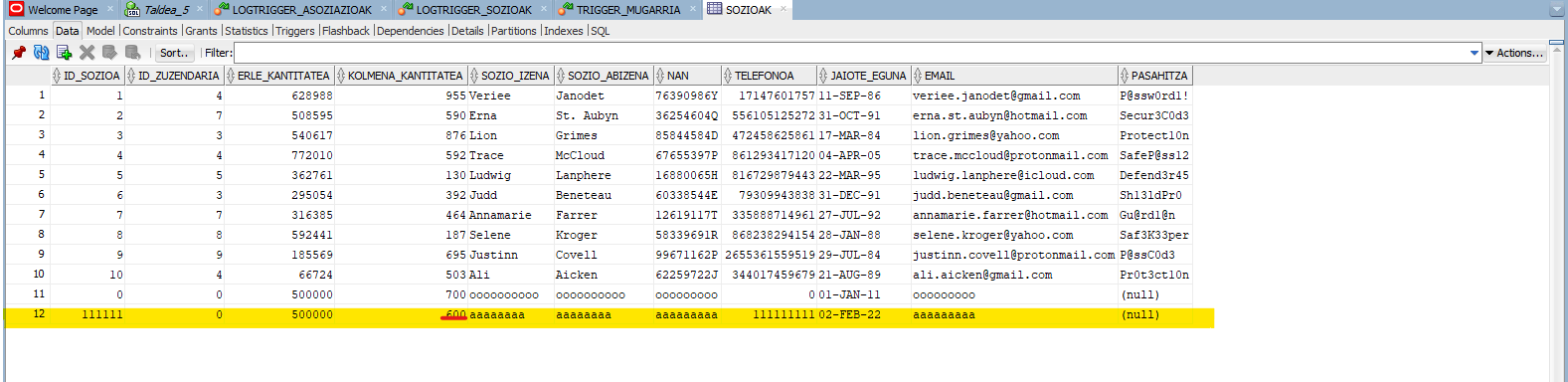
Irudia 4- Sozioa historikoa txertatutako sozioa, log triggerraren bidez.

Beste sozio bat sartu baina erlauntz kantitatea bete ez.

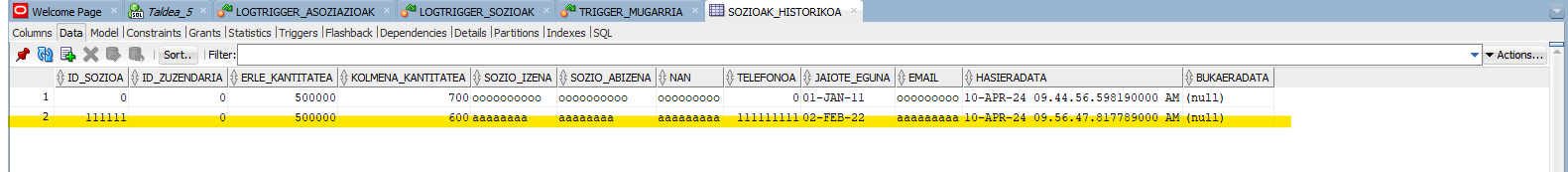


Irudia 5- Erlauntz kantitatea betetzen ez duen sozioa txertatu.

Triggeraren bidez erlauntz kantitatea 100 etik 600 era aldatu da.

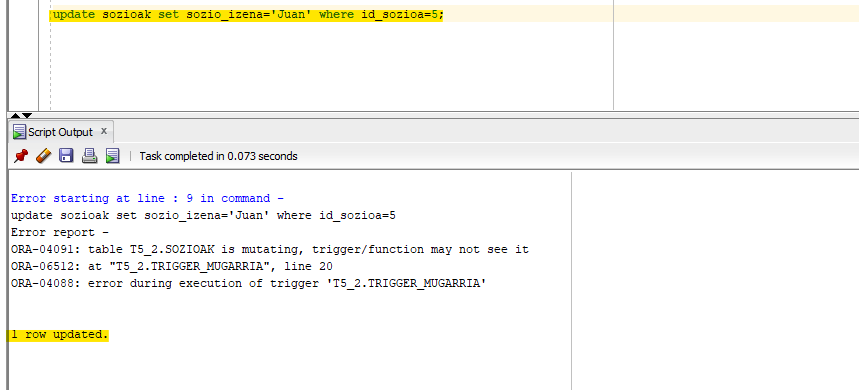


Irudia 6-Sozioa sozioak taulan txertatu eta triggeraren ondorioz eurlantz koprua ezarri.

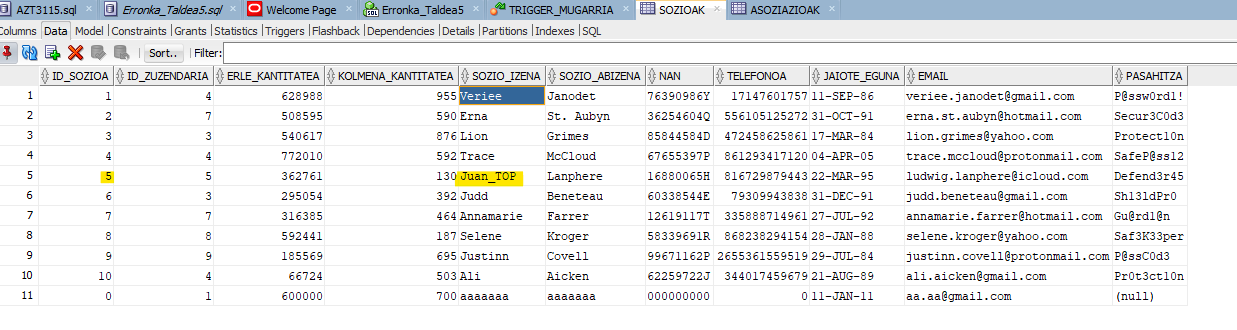


Irudia 7-Sozioak\_historikoa taulan erregistrua log triggerraren ondorioz gehitu.

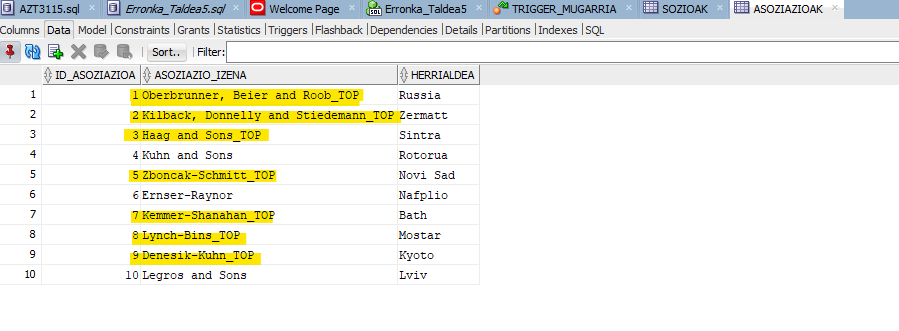
Triggerraren update zihurtatzeko sozio baten izena aldatu zaio. Honek bost produktu salduko ditu berez izena aldatuko zaio baita bere asoziazioarena ere.



Irudia 8- Sozioaren izena aldatu.



Irudia 9-Izena aldatutako sozioaren izen berriari top gehitu.



Irudia 10- Izena aldatutako sozioa parte den asoziazioen izenari top gehitu.