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

Erronka 2.-Sprinta

Aurkibidea:

[Triggerra 6](#_Toc165023898)

[Triggera pausuz pausu 7](#_Toc165023899)

[Log Triggera sortu 12](#_Toc165023900)

[Prozedura 18](#_Toc165023901)

[Funtzioa 19](#_Toc165023902)

[Paketea 20](#_Toc165023903)

[Erabiltzaileak, taldeak eta karpetak kudeatu 24](#_Toc165023904)

[Taldeen sorrera 24](#_Toc165023905)

[Erabiltzaileen sorrera 25](#_Toc165023906)

[Karpeten sorrera 27](#_Toc165023907)

[Karpetei jabea jarri. 28](#_Toc165023908)

[Karpeten baimenak 30](#_Toc165023909)

[Partizioa 31](#_Toc165023910)

[Interfaze grafikoa 34](#_Toc165023911)

[Display nagusia. 34](#_Toc165023912)

[Erregistratu 35](#_Toc165023913)

[LogIn 36](#_Toc165023914)

[Produktuak 41](#_Toc165023915)

Irudien aurkibidea:

[Irudia 1- Erlauntz eta erle kopurua betetzen ez duen sozioa txertatu. 15](#_Toc165031554)

[Irudia 2- Erlauntz eta erle kantitate egokiak dituen sozioa txertatu. 15](#_Toc165031555)

[Irudia 3- Txertatutako sozioa sozioak taulan. 16](#_Toc165031556)

[Irudia 4- Sozioa historikoa txertatutako sozioa, log triggerraren bidez. 16](#_Toc165031557)

[Irudia 5- Erlauntz kantitatea betetzen ez duen sozioa txertatu. 16](#_Toc165031558)

[Irudia 6- Sozioa sozioak taulan txertatu eta triggeraren ondorioz erlauntz kopurua ezarri. 17](#_Toc165031559)

[Irudia 7- Sozioak\_historikoa taulan erregistroa log triggerraren ondorioz gehitu. 17](#_Toc165031560)

[Irudia 8- Sozioaren izena aldatu. 17](#_Toc165031561)

[Irudia 9- Izena aldatutako sozioaren izen berriari top gehitu. 17](#_Toc165031562)

[Irudia 10- Izena aldatutako sozioa parte den asoziazioen izenari top gehitu. 18](#_Toc165031563)

[Irudia 11- Prozedura funtzionamendua. 19](#_Toc165031564)

[Irudia 12- Funtzioa funtzionamendua. 20](#_Toc165031565)

[Irudia 13- Tabla sortuta. 22](#_Toc165031566)

[Irudia 14- Funtzioa eta prozeduraren bloke anonimoa. 24](#_Toc165031567)

[Irudia 15- Taldeak sortu. 25](#_Toc165031568)

[Irudia 16- Makinako taldeak ikusi 25](#_Toc165031569)

[Irudia 17- Erabiltzaileak sortu eta taldeetan sartu 26](#_Toc165031570)

[Irudia 18- Erabiltzaileak ikusi 27](#_Toc165031571)

[Irudia 19- Karpetak sortu 28](#_Toc165031572)

[Irudia 20- Karpeten jabeak 28](#_Toc165031573)

[Irudia 21- Karpeten jabean 29](#_Toc165031574)

[Irudia 22- Karpeten baimenak 30](#_Toc165031575)

[Irudia 23- Diskoa gehitu 31](#_Toc165031576)

[Irudia 24- Diskoa ikusi 31](#_Toc165031577)

[Irudia 25- Diskoa ireki 32](#_Toc165031578)

[Irudia 26- Partizioa egin 32](#_Toc165031579)

[Irudia 27- Partizioa ikusi 33](#_Toc165031580)

[Irudia 28- Fitxategi sistema aldatu 33](#_Toc165031581)

[Irudia 29- Muntaketa puntua eta diskoaren muntaketa 33](#_Toc165031582)

[Irudia 30- Muntaketa puntu iraunkorra 34](#_Toc165031583)

[Irudia 31- Partizioa ikusi 34](#_Toc165031584)

[Irudia 32- Display nagusia. 34](#_Toc165031585)

[Irudia 33- Erregistratu botoia. 35](#_Toc165031586)

[Irudia 34- Erregistratu display-a. 35](#_Toc165031587)

[Irudia 35- Sozioak taula erregistratu aurretik. 36](#_Toc165031588)

[Irudia 36- Erregistratu. 36](#_Toc165031589)

[Irudia 37- Erregistratu ondoren sozioak taula. 36](#_Toc165031590)

[Irudia 38- Login botoia. 37](#_Toc165031591)

[Irudia 39- Login display. 37](#_Toc165031592)

[Irudia 40- Login errorea. 38](#_Toc165031593)

[Irudia 41- Login diplay 2. 38](#_Toc165031594)

[Irudia 42- Logeatu display. 38](#_Toc165031595)

[Irudia 43- Logeatuta display nagusia. 39](#_Toc165031596)

[Irudia 44- Profila. 39](#_Toc165031597)

[Irudia 45- Produktuak botoia. 41](#_Toc165031598)

[Irudia 46- Produktuak menua. 41](#_Toc165031599)

[Irudia 47- Eztiak taula. 42](#_Toc165031600)

[Irudia 48- Materialak taula. 42](#_Toc165031601)

[Irudia 49- Beste batzuk taula. 43](#_Toc165031602)

# 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 pausuz pausu

Lehenengo, kontuan hartu 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 behar da, bata bestearen barruan. Bertan bi baldintza bete behar dira: erlauntz 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, erlauntz kantitatea ez bada betetzen baina erle kantitatea bai, erlauntz kantitatea 600 jarriko da. Horretarako hasierako baldintza beteko balitz beste baldintza bat egon behar da horren barruan. Honek erlauntz 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, azpi kontsultak 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 log 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; |

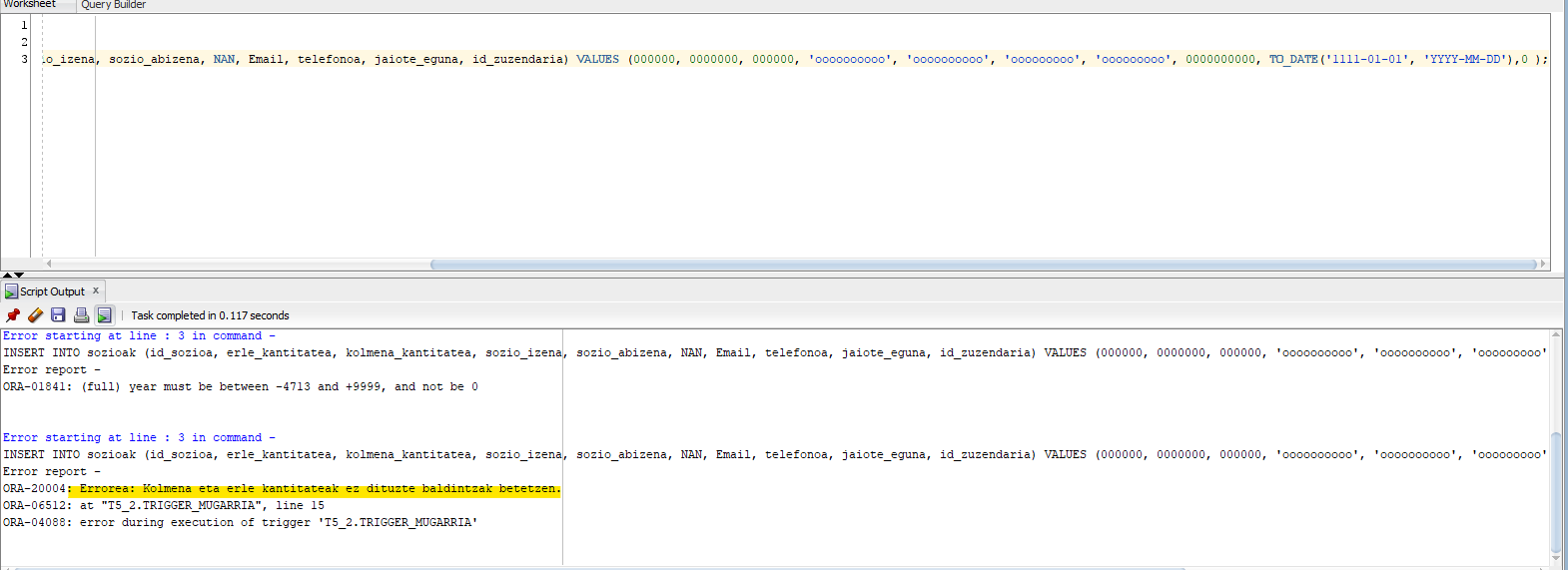
Lehenengo taularen izena aldatu, hemen bi taula kontuan izan behar dugu: aldaketa egiten den taula eta aldaketa horren balio zaharrak eta berriak gordetzeko taula. Lehena sozioak taula izango da eta bertan gertatzen diren aldaketak sozioak\_historikoa taulan agertuko dira bertan aldaketen datak jarriz.

|  |
| --- |
| 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. Aurrekoak 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.

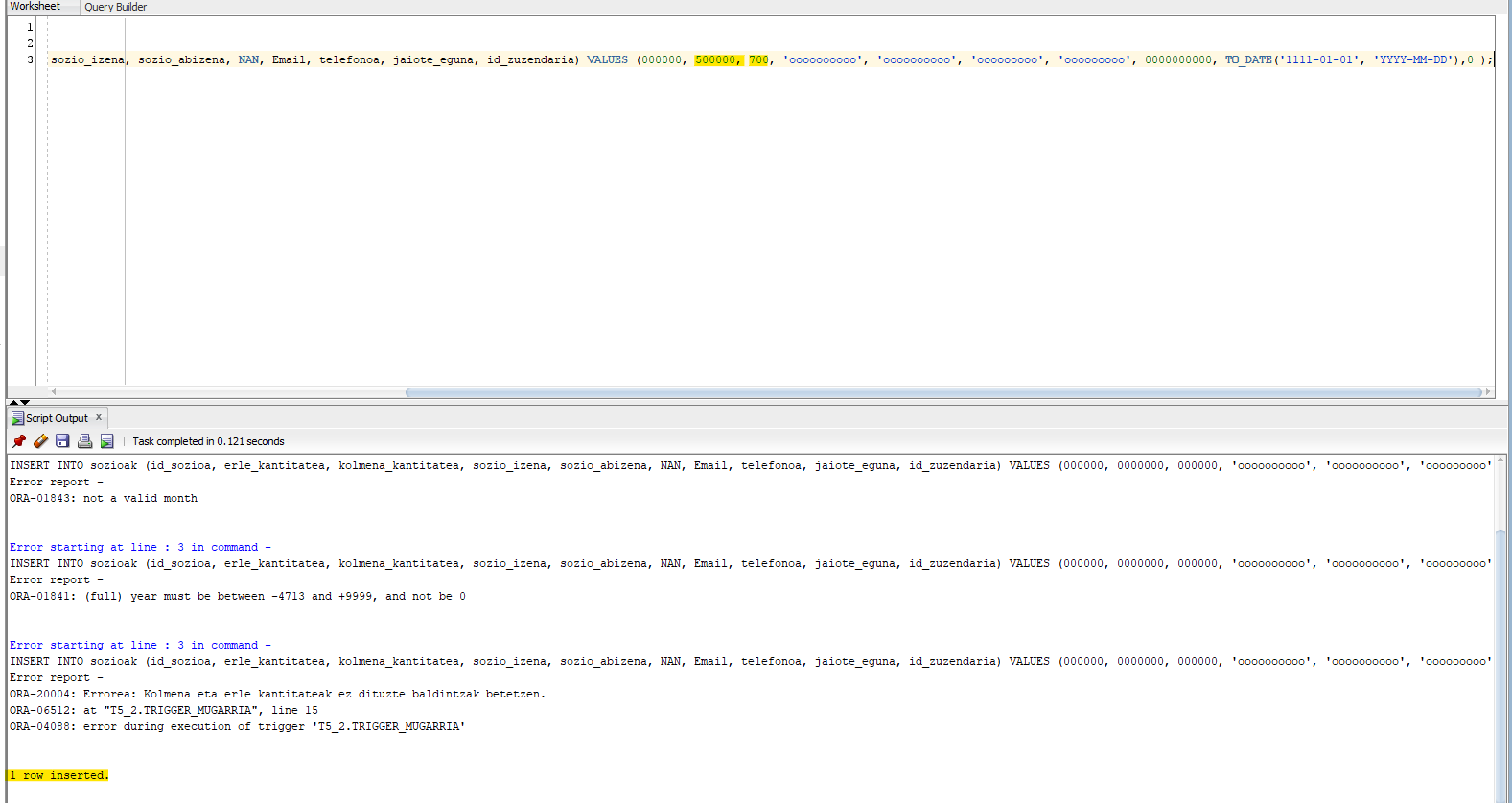
|  |
| --- |
| create or replace NONEDITIONABLE TRIGGER LOGTRIGGER\_ASOZIAZIOAK  AFTER INSERT OR UPDATE OR DELETE ON asoziazioak  FOR EACH ROW  DECLARE Now TIMESTAMP;  BEGIN  SELECT CURRENT\_TIMESTAMP INTO Now FROM Dual;  UPDATE asoziazioak\_historikoa  SET BukaeraData = Now  WHERE BukaeraData IS NULL  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 kantitatea eta erle kantitatea ez badira betetzen, errorea agertzen da.

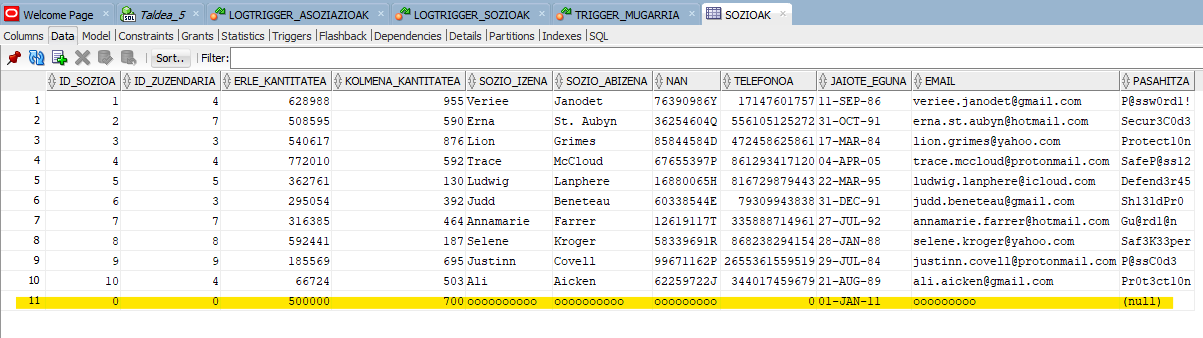


Irudia 1- Erlauntz eta erle kopurua betetzen ez duen sozioa txertatu.

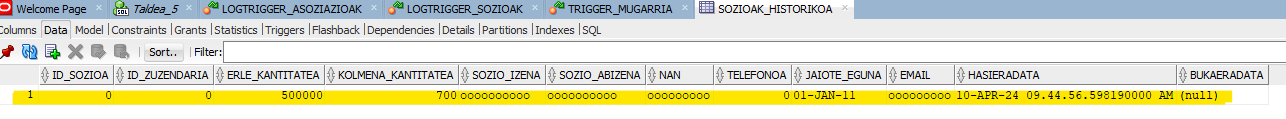
Erlauntz kantitatea eta erle kantitatea lehenengo baldintza betetzen badu baina bigarren ez, hau arazorik gabe txertatuko da, eta sozioak historikoan ere ongi txertatuko da.



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

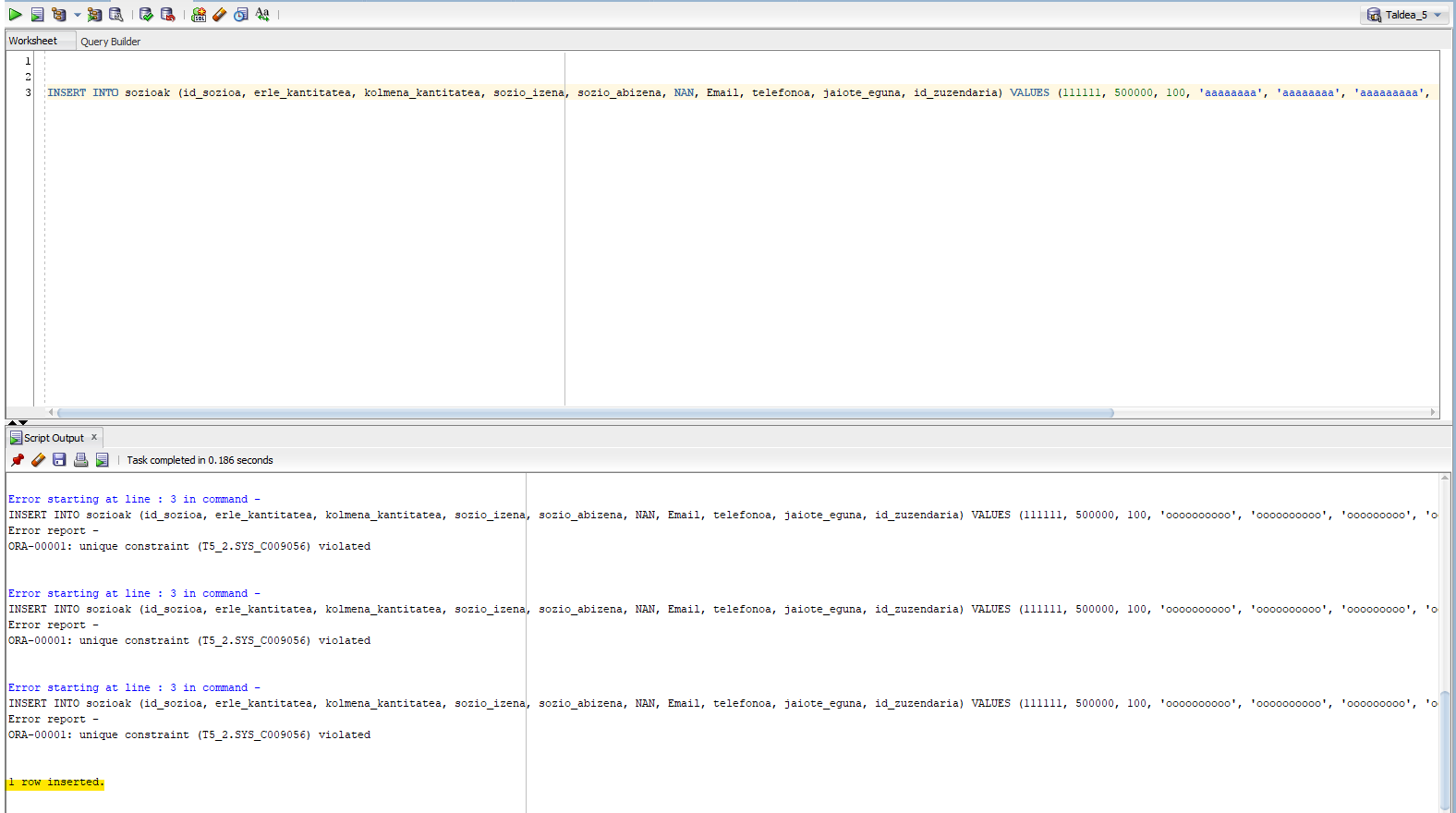


Irudia 3- Txertatutako sozioa sozioak taulan.



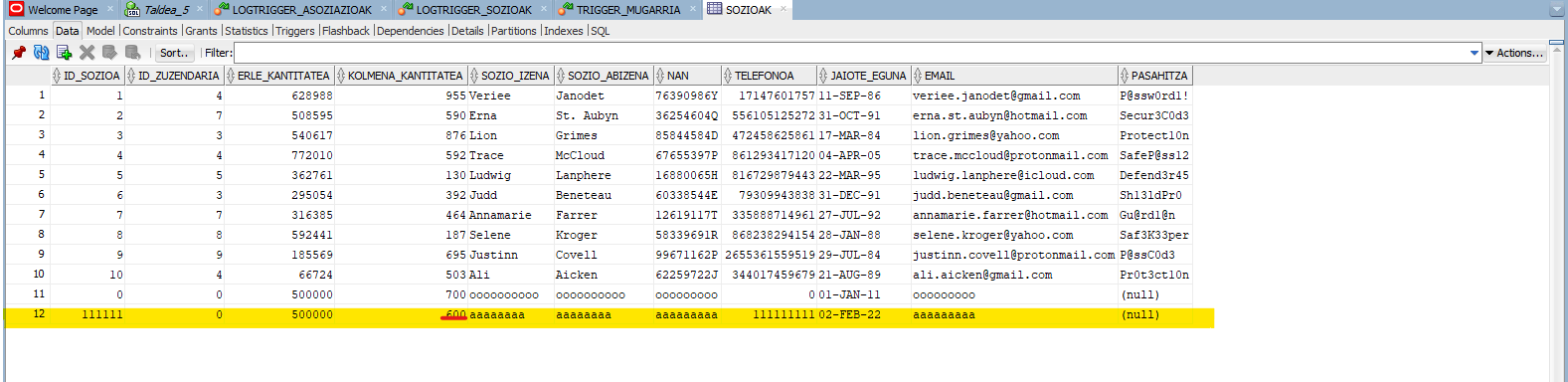
Irudia 4- Sozioa historikoa txertatutako sozioa, log triggerraren bidez.

Beste sozio bat sartu baina erlauntz kantitatea bete ez.

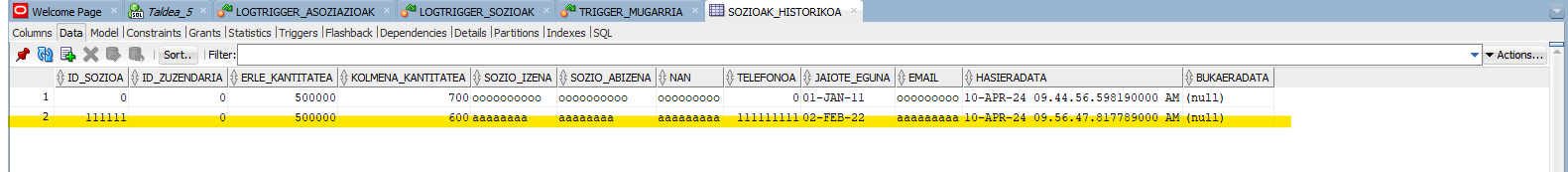


Irudia 5- Erlauntz kantitatea betetzen ez duen sozioa txertatu.

Triggeraren bidez erlauntz kantitatea 100 etik 600 era aldatu da.

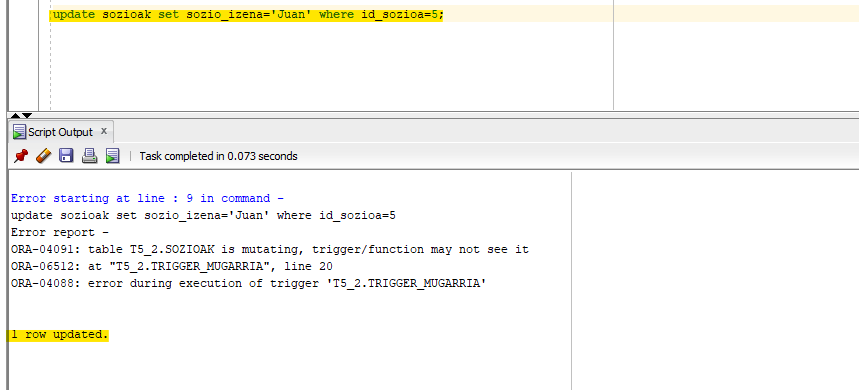


Irudia 6- Sozioa sozioak taulan txertatu eta triggeraren ondorioz erlauntz kopurua ezarri.



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

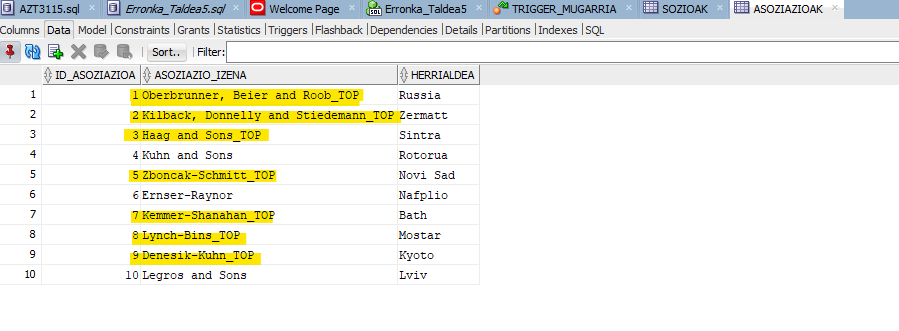
Triggerraren update ziurtatzeko sozio baten izena aldatu zaio. Honek bost produktu salduko ditu, berez izena aldatuko zaio baita bere asoziaziorena ere.



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.

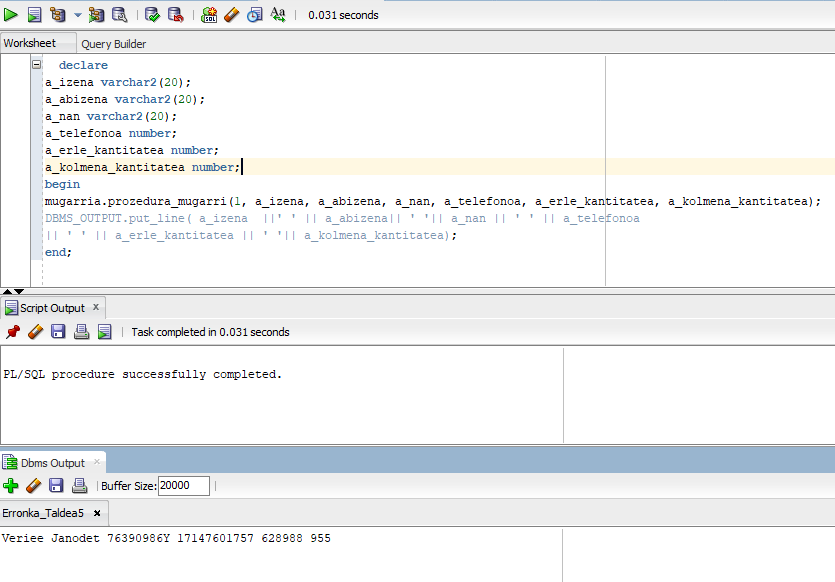
# Prozedura

Prozedura bat sortu da, honek id\_sozioa jasota, sozioaren datu guztiak itzuliko ditu.

|  |
| --- |
| create or replace PROCEDURE Prozedura\_mugarri  (  a\_id\_sozioa IN NUMBER  , a\_izena OUT varchar2  , a\_abizena OUT varchar2  , a\_NAN OUT varchar2  , a\_telefonoa OUT number  , a\_erle\_kantitatea OUT number  , a\_kolmena\_kantitatea OUT number  ) AS  begin  select sozio\_izena, sozio\_abizena, nan, telefonoa, erle\_kantitatea, kolmena\_kantitatea  into a\_izena, a\_abizena, a\_NAN, a\_telefonoa, a\_erle\_kantitatea, a\_kolmena\_kantitatea from sozioak where a\_id\_sozioa= id\_sozioa;  end Prozedura\_mugarri; |

Honek funtzionatzen duela ziurtatzeko bloke anonimo bat egin da

|  |
| --- |
| declare  a\_izena varchar2(20);  a\_abizena varchar2(20);  a\_nan varchar2(20);  a\_telefonoa number;  a\_erle\_kantitatea number;  a\_kolmena\_kantitatea number;  begin  mugarria.prozedura\_mugarri(1, a\_izena, a\_abizena, a\_nan, a\_telefonoa, a\_erle\_kantitatea, a\_kolmena\_kantitatea);  DBMS\_OUTPUT.put\_line( a\_izena ||' ' || a\_abizena|| ' '|| a\_nan || ' ' || a\_telefonoa  || ' ' || a\_erle\_kantitatea || ' '|| a\_kolmena\_kantitatea);  end |



Irudia 11- Prozedura funtzionamendua.

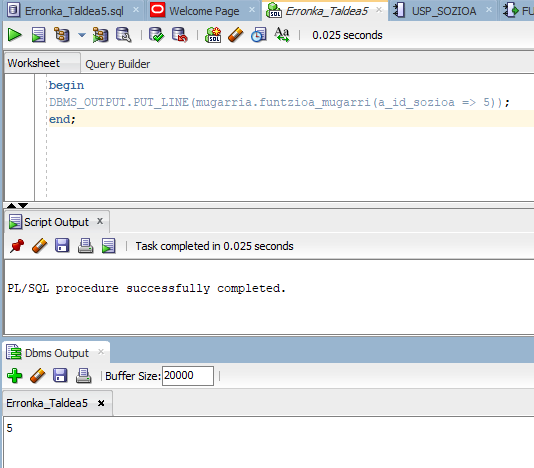
# Funtzioa

Prozedura sortu ondoren funtzioa garatu da. Funtzioak id\_sozioa satu eta honen kolmena kantitatea 500 baino handiagoa bada sozio hori parte den asoziazio kopurua itzuliko du. Baina baldintza ez badu betetzen honek produzitutako ezti mota kantitatea itzuliko du.

|  |
| --- |
| create or replace NONEDITIONABLE FUNCTION Funtzioa\_mugarri  (  A\_ID\_SOZIOA IN NUMBER  ) RETURN VARCHAR2 AS  a\_parte number;  a\_kop number;  a\_mota number;  BEGIN  select kolmena\_kantitatea into a\_kop from sozioak where id\_sozioa= a\_id\_sozioa;  if a\_kop>500 then  select count(id\_asoziazioa) into a\_parte from asoziazio\_parte where a\_id\_sozioa=id\_sozioa;  RETURN a\_parte;    else  select count(id\_produktua) into a\_mota from ze\_ezti where a\_id\_sozioa=id\_sozioa;  RETURN a\_mota;  end if;  END Funtzioa\_mugarri; |

Prozeduraren moduan hau iurtatzeko bloke anonimo bat sortu da.

|  |
| --- |
| begin  DBMS\_OUTPUT.PUT\_LINE(mugarria.funtzioa\_mugarri(a\_id\_sozioa => 5));  end |



Irudia 12- Funtzioa funtzionamendua.

# Paketea

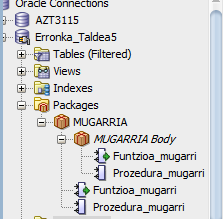
Azkenik prozedura eta funtzioa sortuta, pakete bat sortu da bertan funtzioa eta prozedura edukitzeko.

Horretako lehenik paketea sortu, bertan funtzioak eta prozedurak edukiko dituzten aldagaiak definituz.

|  |
| --- |
| CREATE OR REPLACE  PACKAGE MUGARRIA AS  PROCEDURE Prozedura\_mugarri  (  a\_id\_sozioa IN NUMBER  , a\_izena OUT varchar2  , a\_abizena OUT varchar2  , a\_NAN OUT varchar2  , a\_telefonoa OUT number  , a\_erle\_kantitatea OUT number  , a\_kolmena\_kantitatea OUT number  );  FUNCTION Funtzioa\_mugarri  (  A\_ID\_SOZIOA IN NUMBER  ) RETURN VARCHAR2;  /\* TODO enter package declarations (types, exceptions, methods etc) here \*/  END MUGARRIA; |

Ondoren paketearen gorputza sortu behar da. Bertan aurretik sortuta zeuden funtzioa eta prozedura inplementatuko dira.

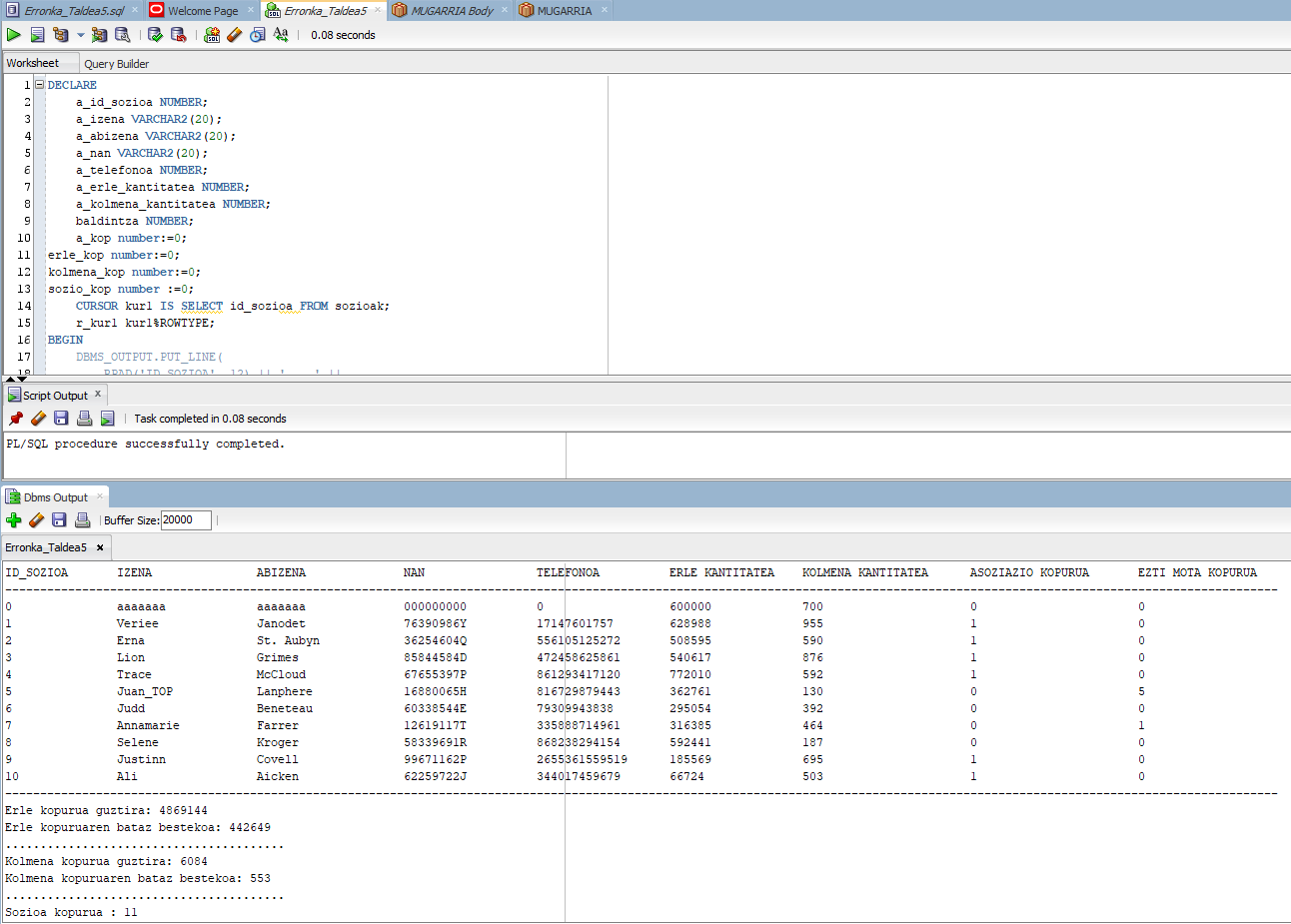
|  |
| --- |
| create or replace NONEDITIONABLE PACKAGE BODY Mugarria AS  -- Implementación del procedimiento  PROCEDURE Prozedura\_mugarri  (  a\_id\_sozioa IN NUMBER  , a\_izena OUT varchar2  , a\_abizena OUT varchar2  , a\_NAN OUT varchar2  , a\_telefonoa OUT number  , a\_erle\_kantitatea OUT number  , a\_kolmena\_kantitatea OUT number  )AS  begin  select sozio\_izena, sozio\_abizena, nan, telefonoa, erle\_kantitatea, kolmena\_kantitatea  into a\_izena, a\_abizena, a\_NAN, a\_telefonoa, a\_erle\_kantitatea, a\_kolmena\_kantitatea from sozioak where a\_id\_sozioa= id\_sozioa;  end Prozedura\_mugarri;    -- Implementación de la función  FUNCTION Funtzioa\_mugarri  (  A\_ID\_SOZIOA IN NUMBER  ) RETURN VARCHAR2 AS  a\_parte number;  a\_kop number;  a\_mota number;  BEGIN  select kolmena\_kantitatea into a\_kop from sozioak where id\_sozioa= a\_id\_sozioa;  if a\_kop>500 then  select count(id\_asoziazioa) into a\_parte from asoziazio\_parte where a\_id\_sozioa=id\_sozioa;  RETURN a\_parte;    else  select count(id\_produktua) into a\_mota from ze\_ezti where a\_id\_sozioa=id\_sozioa;  RETURN a\_mota;  end if;  END Funtzioa\_mugarri;  END Mugarria |



Irudia 13- Tabla sortuta.

Azkenik honek guztiak funtzionatzen duela ziurtatzeko bloke anonimo bat sortu da. Honek sozio guztien datuak eta ezti mota kopurua edo asoziazio kopurua bistaratuko du.

|  |
| --- |
| DECLARE  a\_id\_sozioa NUMBER;  a\_izena VARCHAR2(20);  a\_abizena VARCHAR2(20);  a\_nan VARCHAR2(20);  a\_telefonoa NUMBER;  a\_erle\_kantitatea NUMBER;  a\_kolmena\_kantitatea NUMBER;  baldintza NUMBER;  a\_kop number:=0;  erle\_kop number:=0;  kolmena\_kop number:=0;  sozio\_kop number :=0;  CURSOR kur1 IS SELECT id\_sozioa FROM sozioak;  r\_kur1 kur1%ROWTYPE;  BEGIN  DBMS\_OUTPUT.PUT\_LINE(  RPAD('ID\_SOZIOA', 12) || ' ' ||  RPAD('IZENA', 16) || ' ' ||  RPAD('ABIZENA', 17) || ' ' ||  RPAD('NAN', 15) || ' ' ||  RPAD('TELEFONOA', 15) || ' ' ||  RPAD('ERLE KANTITATEA', 15) || ' ' ||  RPAD('KOLMENA KANTITATEA', 20)|| ' ' ||  RPAD('ASOZIAZIO KOPURUA', 20) || ' ' ||  RPAD('EZTI MOTA KOPURUA', 20)  );  DBMS\_OUTPUT.PUT\_LINE('--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------');  OPEN kur1;  FETCH kur1 INTO r\_kur1;  WHILE kur1%FOUND LOOP  mugarria.prozedura\_mugarri(r\_kur1.id\_sozioa, a\_izena, a\_abizena, a\_nan, a\_telefonoa, a\_erle\_kantitatea, a\_kolmena\_kantitatea);  baldintza:= mugarria.funtzioa\_mugarri(a\_id\_sozioa => r\_kur1.id\_sozioa);  select kolmena\_kantitatea into a\_kop from sozioak where id\_sozioa=r\_kur1.id\_sozioa;    if a\_kop >500 then  DBMS\_OUTPUT.PUT\_LINE(  RPAD(r\_kur1.id\_sozioa, 12) || ' ' ||  RPAD(a\_izena, 13) || ' ' ||  RPAD(a\_abizena, 18) || ' ' ||  RPAD(a\_nan, 12) || ' ' ||  RPAD(a\_telefonoa, 14) || ' ' ||  RPAD(a\_erle\_kantitatea, 12) || ' ' ||  RPAD(a\_kolmena\_kantitatea, 17) || ' '||  RPAD(baldintza, 17) || ' ' ||  RPAD('0', 12)  );  else  DBMS\_OUTPUT.PUT\_LINE(  RPAD(r\_kur1.id\_sozioa, 12) || ' ' ||  RPAD(a\_izena, 13) || ' ' ||  RPAD(a\_abizena, 18) || ' ' ||  RPAD(a\_nan, 12) || ' ' ||  RPAD(a\_telefonoa, 14) || ' ' ||  RPAD(a\_erle\_kantitatea, 12) || ' ' ||  RPAD(a\_kolmena\_kantitatea, 17) || ' '||  RPAD('0', 17) || ' ' ||  RPAD(baldintza, 12)  );  end if;  erle\_kop:=erle\_kop+a\_erle\_kantitatea;  kolmena\_kop:=kolmena\_kop+a\_kolmena\_kantitatea;  sozio\_kop:= sozio\_kop+1;  FETCH kur1 INTO r\_kur1;  END LOOP;  CLOSE kur1;  DBMS\_OUTPUT.PUT\_LINE('--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------');  DBMS\_OUTPUT.PUT\_LINE('Erle kopurua guztira: '||erle\_kop);  DBMS\_OUTPUT.PUT\_LINE('Erle kopuruaren bataz bestekoa: ' ||round((erle\_kop/sozio\_kop)));  DBMS\_OUTPUT.PUT\_LINE('........................................');  DBMS\_OUTPUT.PUT\_LINE('Kolmena kopurua guztira: '||kolmena\_kop);  DBMS\_OUTPUT.PUT\_LINE('Kolmena kopuruaren bataz bestekoa: ' ||round((kolmena\_kop/sozio\_kop)));  DBMS\_OUTPUT.PUT\_LINE('........................................');    DBMS\_OUTPUT.PUT\_LINE('Sozioa kopurua : '||sozio\_kop);  END;  / |

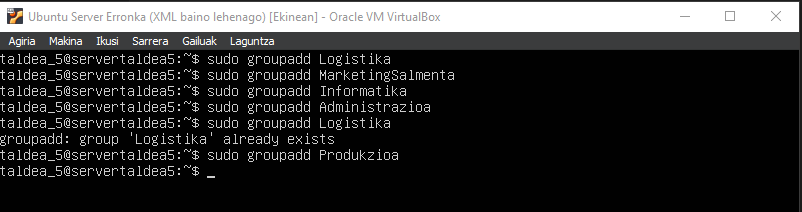


Irudia 14- Funtzioa eta prozeduraren bloke anonimoa.

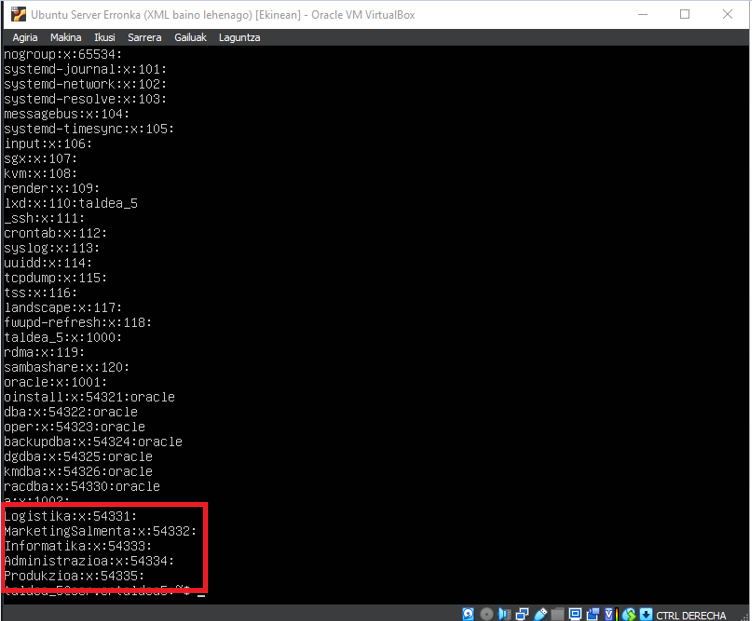
## Erabiltzaileak, taldeak eta karpetak kudeatu

## Taldeen sorrera

Terminal bidez taldeak sortzeko hurrengoko komandoa erabili da: **sudo groupadd <talde\_izena>.** Horrekin bost taldeak sortu dira.



Irudia 15- Taldeak sortu.

Ondoren **cat /etc/group** komandoa erabiliz sortutako taldeak sortu diren egiaztatu da. 

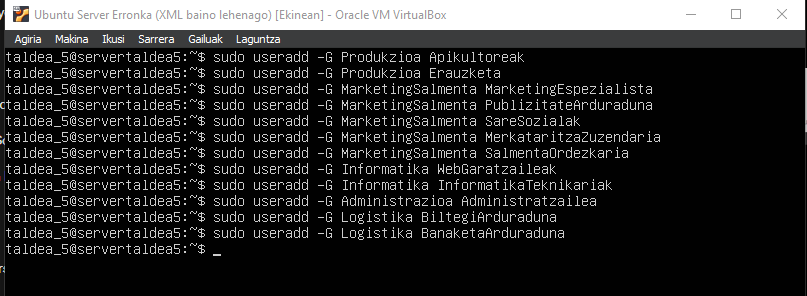
Irudia 16- Makinako taldeak ikusi

## Erabiltzaileen sorrera

Enpresako lankide bakoitzeko erabiltzaile bat sortuko da. Erabiltzailea sortzerakoan bakoitza bere departamentuari dagokion taldean sartuko da.

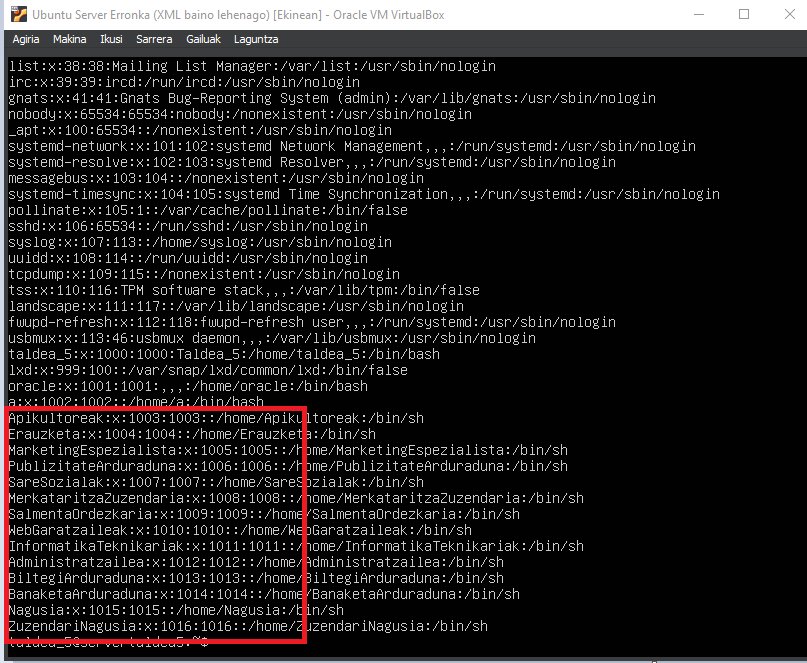
* Produkzioa taldea:
  + Apikultoreak
  + Erauzketa
* MarketingSalmenta taldea:
  + MarketingEzpezialista
  + SareSozialak
  + MerkataritzaZuzendaria
  + SalmentaOrdezkaria
* Irformatika taldea:
  + WebGaratzaileak
  + InformatikaGaratzaileak
* Logistika taldea:
  + BiltegiArduraduna
  + BanaketaArduraduna

Talde guztiak sortu ondoren. Erabiltzaileak sortu eta talde bakoitzean sartu behar dira. Horretarako dena komando bakarra erabili da: **sudo useradd -G <talde\_izena> <erabiltzaile\_berria>.** Horrela erabiltzailea sortu eta zuzenean sartuko da taldean.



Irudia 17- Erabiltzaileak sortu eta taldeetan sartu

Erabiltzaile guztiak sortu ondoren denak ongi sortu diren jakiteko **cat /etc/passwd** erabili da.



Irudia 18- Erabiltzaileak ikusi

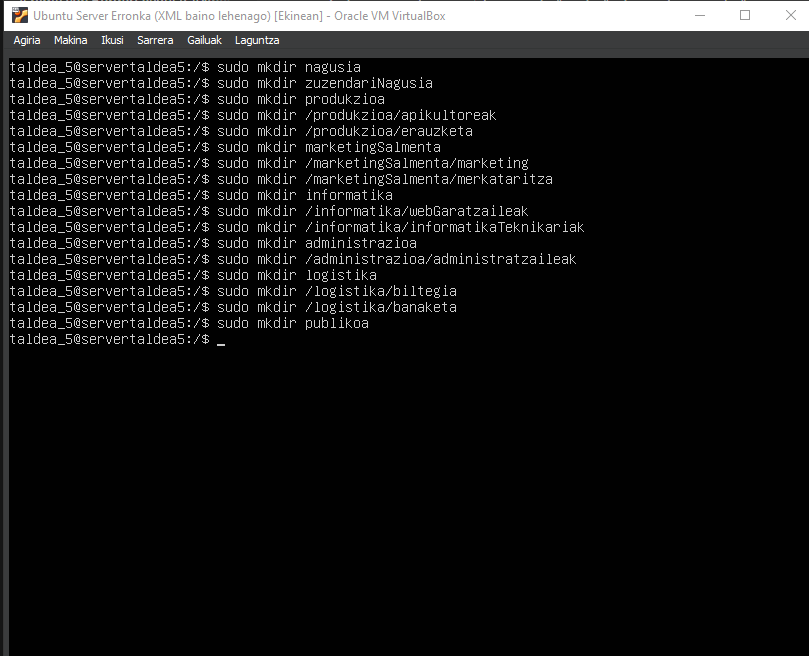
## Karpeten sorrera

Dokumentuan ageri den moduan enpresako erabiltzaile bakoitzak bere karpeta izango du. Baita enpresako talde bakoitzak ere. Horretaz gain publikoa izeneko karpeta bat egongo da enpresa osoko erabiltzaileentzat.

Lau karpeta mota ezberdin daude:

* Karpeta pribatuak: nagusia eta zuzendari nagusiaren karpetak izango dira.
* Taldeko karpetak: Departamentu bakoitzak karpeta bat edukiko du.
* Erabiltzaileen karpetak: Departamentu bakoitzeko erabiltzaile bakoitzak edukiko duen karpetak izango dira.
* Karpeta publikoa: Enpresa osoan karpeta bat egongo da erabiltzaile guztientzat.

Karpetak sortzeko **sudo mkdir <karpeta\_izena>** komandoa erabili da.

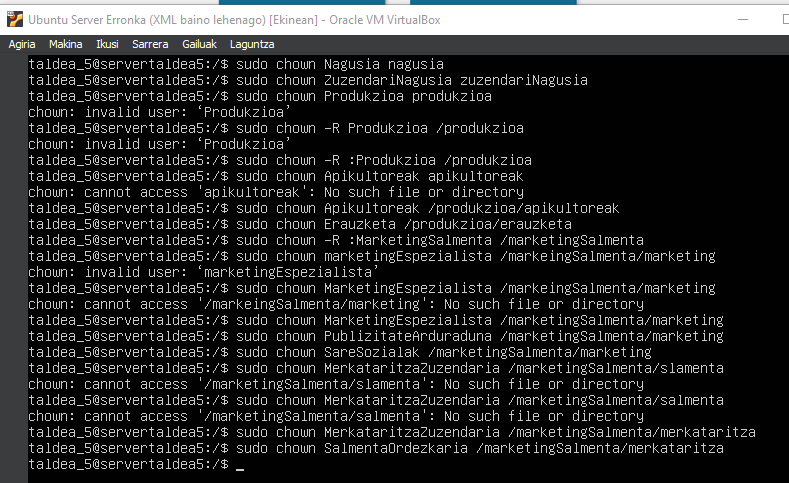


Irudia 19- Karpetak sortu

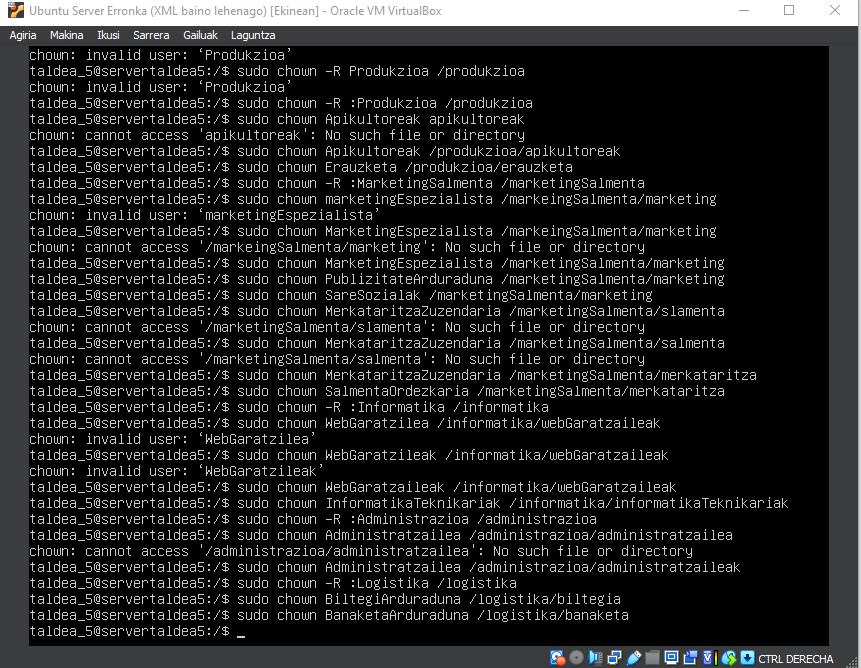
## Karpetei jabea jarri.

Erabiltzaile bakoitza bere karpetaren jabea izango da. Departamentu bakoitzeko karpetako jabea taldeak izango dira.

Erabiltzaile bakoitzeko karpetei jabea **sudo chown <erabiltzailea> <karpeta>** komandoa erabili da. Bestalde taldeko karpetei jabe moduan talde bat jartzeko **sudo chown –R :<talde\_izena> <karpeta> .**



Irudia 20- Karpeten jabeak



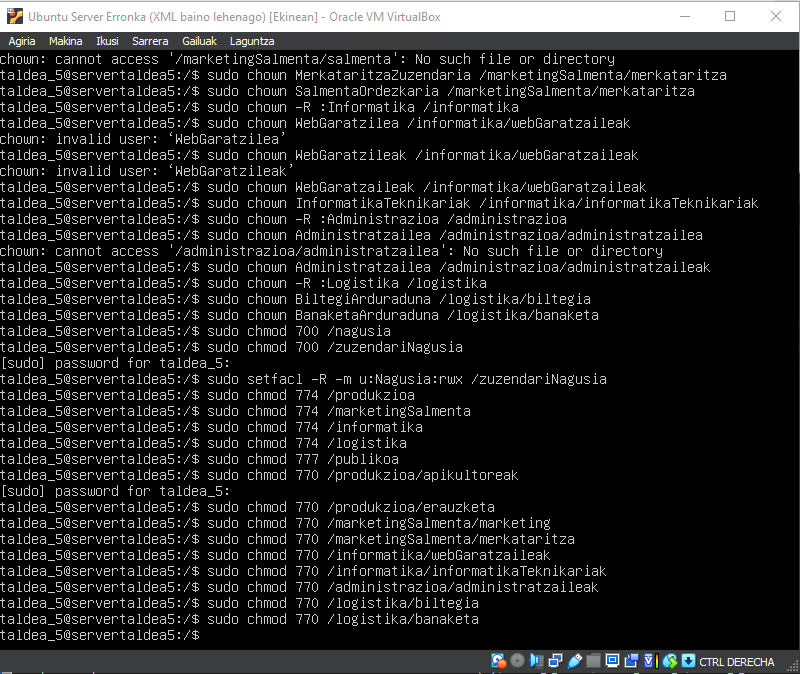
Irudia 21- Karpeten jabean

## Karpeten baimenak

Karpeta bakoitzari baimenak eman zaie. Horretarako **sudo chmod xxx <karpeta>** komandoa erabili da.

Karpeta bakoitzaren baimenen arabera zenbaki ezberdina edukiko dute.

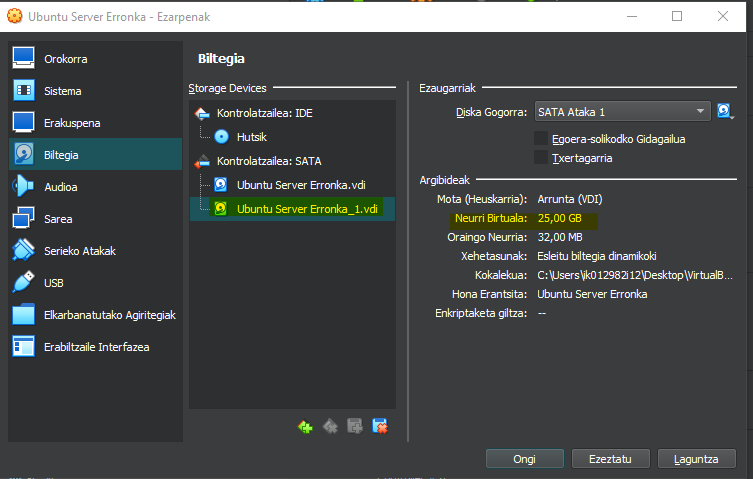
* 770 duten kasuan erabiltzaileak eta taldeak hiru baimenak edukiko dituzte, karpeta honek erabiltzaileen karpetak izango dira.
* 774 ko kasuan taldeen karpetak edukiko dute erabiltzaileak eta taldekoek hiru baimenak eta beste edozeinek ikusteko aukera bakarrik izango dute .
* 777 duen karpetak, publikoak izango dira eta enpresa osoko erabiltzaile guztiak hiru baimenak edukiko dituzte.
* 700 baimenak duten karpetak egongo dira, honek pribatuak direnez jabeak edukiko du karpetako baimen guztiak.



Irudia 22- Karpeten baimenak

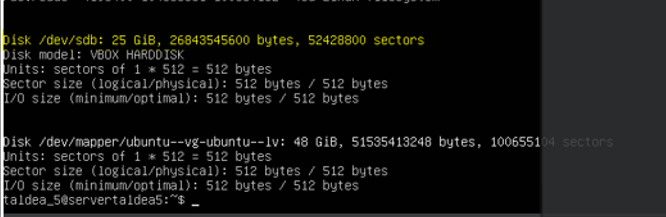
## Partizioa

Partizioa egiteko disko bat gehitu zaio makina birtualari 25gb-koa

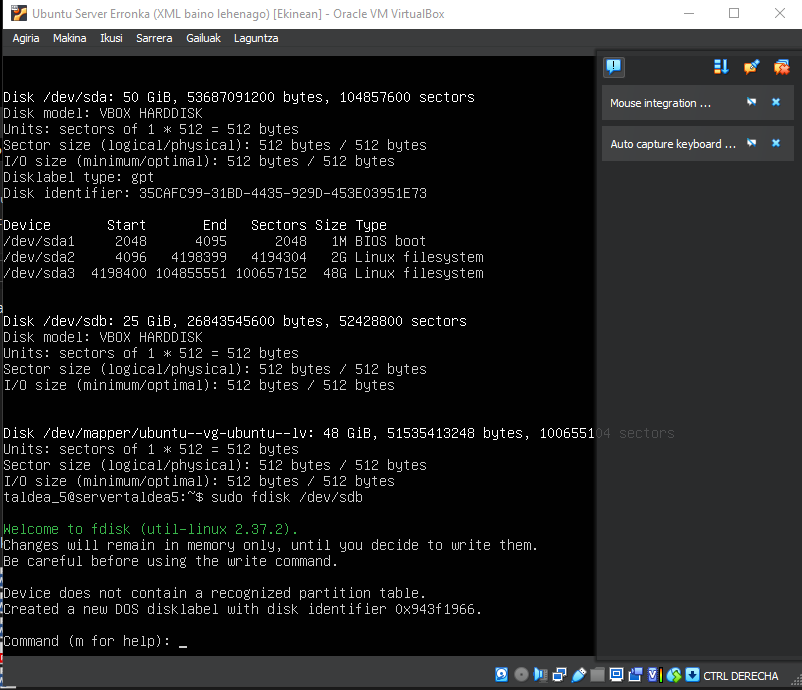


Irudia 23- Diskoa gehitu

**Sudo fdisk –l** komandoarekin sortutako diskoa ongi sortu dela ziurtatu da.

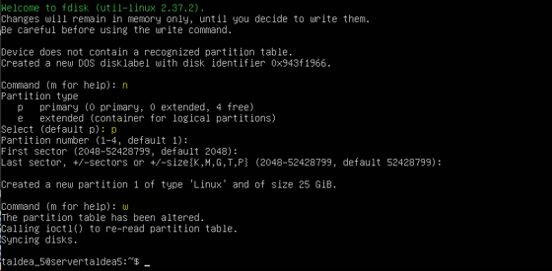


Irudia 24- Diskoa ikusi

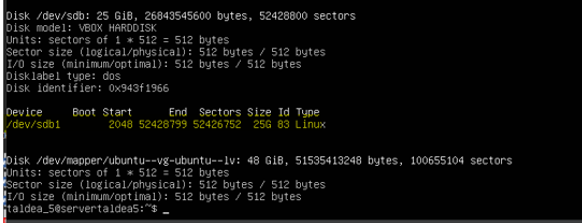
Partizioa egiteko diskoa ireki **sudo fdisk /dev/sdb** komandoa erabiliz.

Irudia 25- Diskoa ireki

Diskoa irekita partizioa egin behar da horretarako **n** komandoa erabili, ondoren “primary “motako partizioa hautatu **a** komando bidez eta hurrengoko aukera guztiak “default” jarri. Azkenik **w** komando exekutatu partizioa gordetzeko.

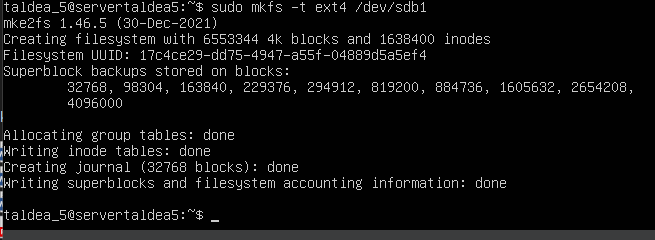


Irudia 26- Partizioa egin

**Sudo fdisk –l** komandoa erabili partizioa ikusteko. 

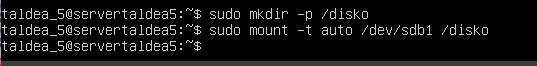
Irudia 27- Partizioa ikusi

Partizioa sortuta fitxategi sistema aldatu behar zaio **sudo mkfs –t ext4 /dev/sdb1**  komadoa erabiliz.



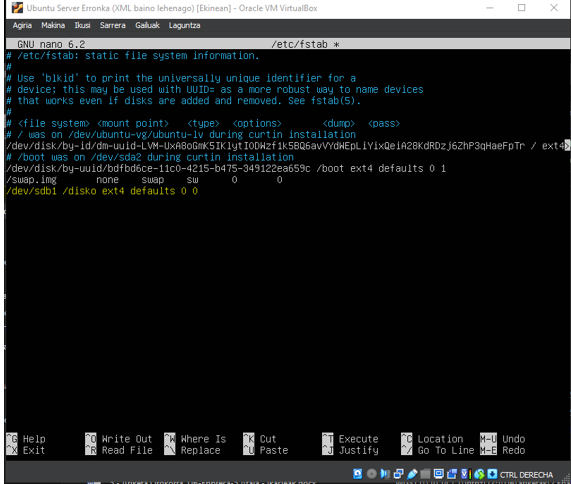
Irudia 28- Fitxategi sistema aldatu

Muntaketa puntua sortu **sudo mkdir –p /disko** komandoarekin eta partizioa montatu **sudo mount –t auto/dev/sdb1 /disko**  komandoarekin.



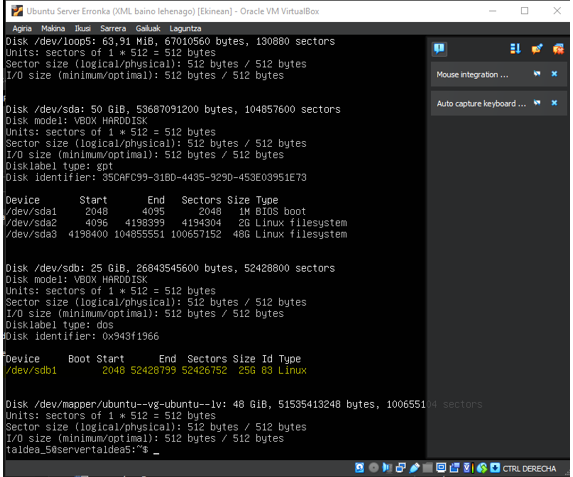
Irudia 29- Muntaketa puntua eta diskoaren muntaketa

Azkenik muntaketa puntua iraunkorra egiteko **sudo nano /etc/fstab** exekutatu eta **/dev/sdb1 /disko ext4 defaults 0 0** gehitu.



Irudia 30- Muntaketa puntu iraunkorra

Bukatzeko **Sudo fdisk –l** komandoa erabili dena ongi atera dela ziurtatzeko.

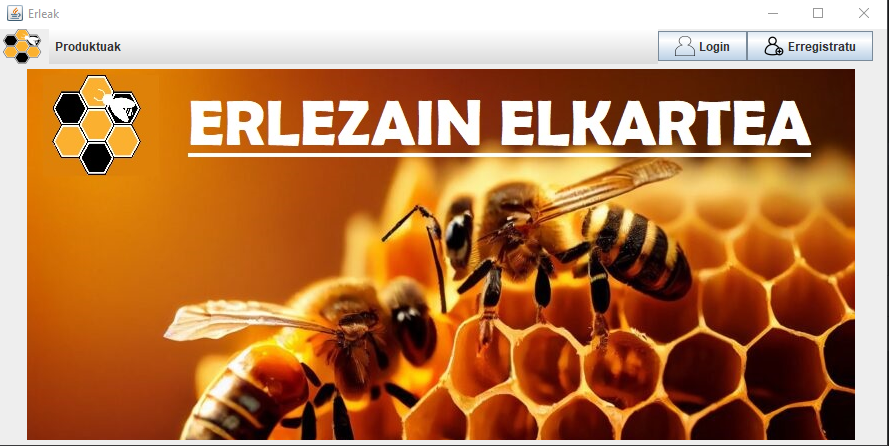


Irudia 31- Partizioa ikusi

# Interfaze grafikoa

## Display nagusia.

Aplikazioan sartzerakoan hurrengoko interfazea pantailaratuko da. Honek ainbat funtzio edukiko ditu.



Irudia 32- Display nagusia.

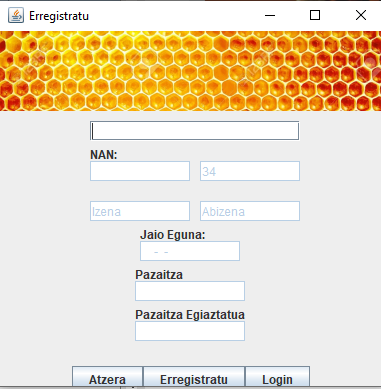
## Erregistratu

Alde batetik **erregistratu** botoia egongo da.



Irudia 33- Erregistratu botoia.

Bertan klikatzerakoan beste display bat irekiko da bertan pertsonaren datuak sartzeko aukerarekin.

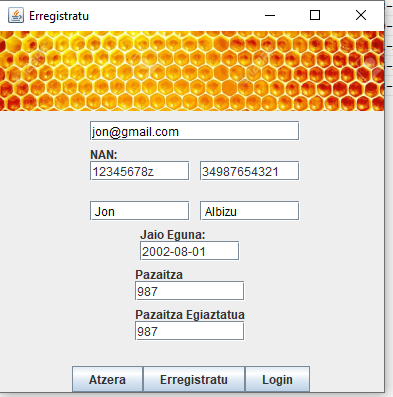


Irudia 34- Erregistratu display-a.

Hemen sozioaren datuak sartu ahalko dira eta erregistratu botoiari eman . Hau aurretik erregistratu ez den sozio batek egin beharko luke. Datuak sartzen badira eta erregistratu botoiari heman, datu basean sozio bat gehituko da.



Irudia 35- Sozioak taula erregistratu aurretik.



Irudia 36- Erregistratu.



Irudia 37- Erregistratu ondoren sozioak taula.

## LogIn

Display-an dagoen beste botoia login botoia izango da.



Irudia 38- Login botoia.

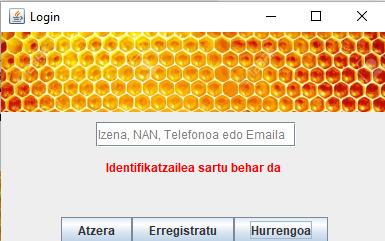
Botoiari klik eginez gero beste display bat irekiko da, non gure izena, nan-a, telefonoa edo email-a eskatuko den.

Bertan datua sartu ondoren hurrengoa botoiari eman eta pasahitza eskatuko da. Kasu honetan nan zenbakiarekin logeatuko da.



Irudia 39- Login display.

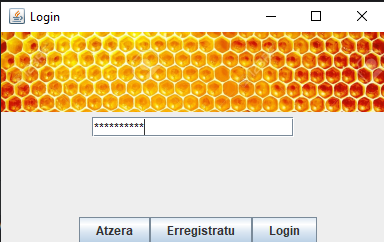
Baina ez bada identifikatzailerik sartzen ezingo da haurrera jarraitu.



Irudia 40- Login errorea.

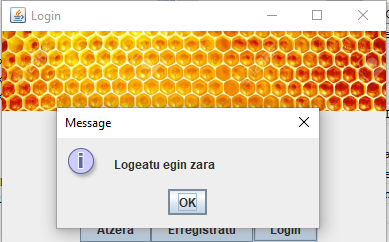
Pasahitz okerra sartuz gero login botoiak ez du ezer egingo beraz ezingo da logeatu.

Bestalde pasahitz zuzena sartuz gero sozioa logeatuko da.



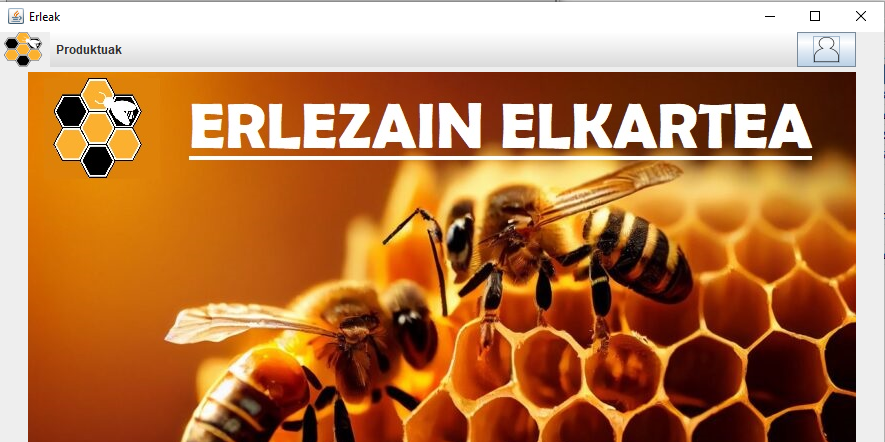
Irudia 41- Login diplay 2.

Hau sartu ondoren logeatu botoiari eman eta sozioa logeatuko da. Bestalde atzera botoiarekin aurreko display-era joan al Azkenik erregistratu botoiarekin, erregistratzeko orria irekiko da.



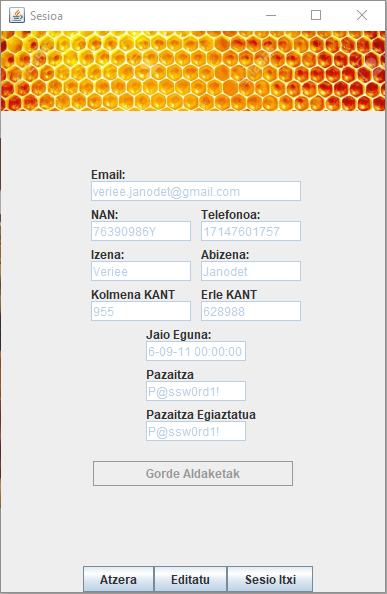
Irudia 42- Logeatu display.

Logeatu ondoren display nagusian profilaren botoia agertuko da.



Irudia 43- Logeatuta display nagusia.

Bertan klik egiten bada gure logeatutako sozioaren datuak pantailaratuko dira.

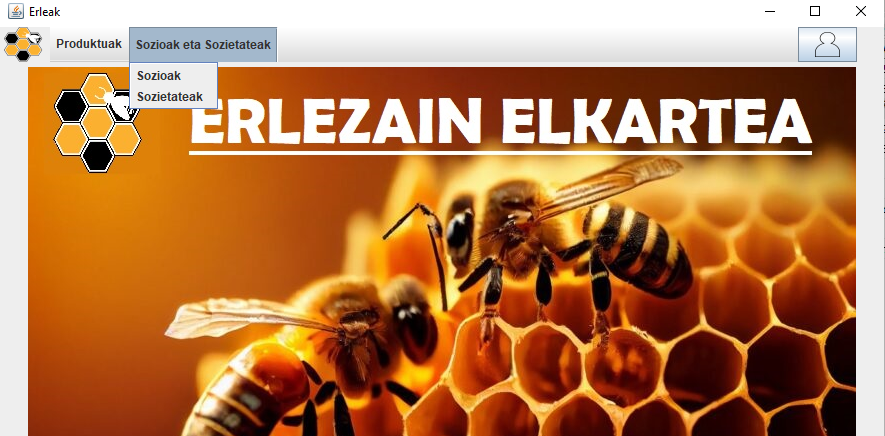


Irudia 44- Profila.

Profilean editatu botoiari emanez gero datuak editatzeko aukera egongo da.

Bestalde sesio itxi botoiari klik eginez gero irekitako sesioa itxiko da eta aurreko display nagusia ikusiko da. Azkenik atzera botoiarekin aurreko display-era bidaliko du.

Logetutako sozioa zuzendaria bada, display nagusian beste menu botoi bat gehituko sozioak eta sozietateak izenekoak.



Sozioetan klik egiten bada, bertan zuzendaria den sozioak sozioen informazioa bistaratuko da.



Eta sozietateak botoian klik egiten bada bertan sozietate guztien informazioa bistaratuko da. Honetaz gain logoaren gainean klikatzen badugu display nagusira joango da.



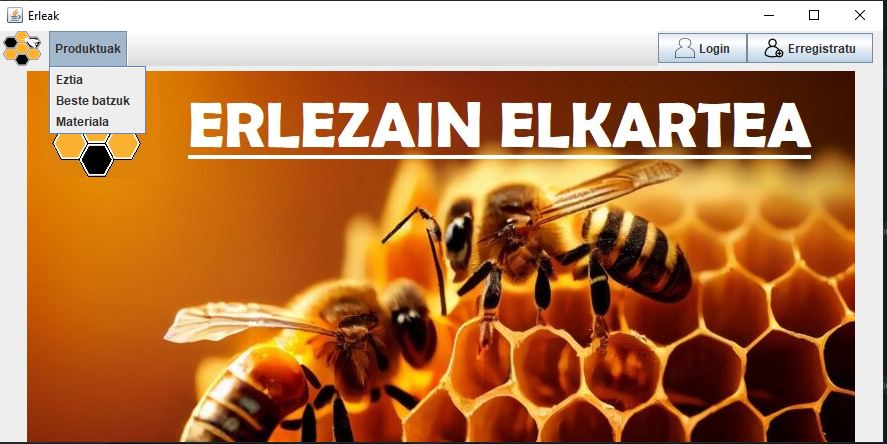
Horretaz gain sozioa logeatuta badago eta programa ixten badu, sozioaren datuan dokumentu batean gordeko dira eta berriro programa abiatzen bada, sozioak logeatuta jarraituko du.

## Produktuak

Azkenik produktuak botoia ezango du.

Irudia 45- Produktuak botoia.

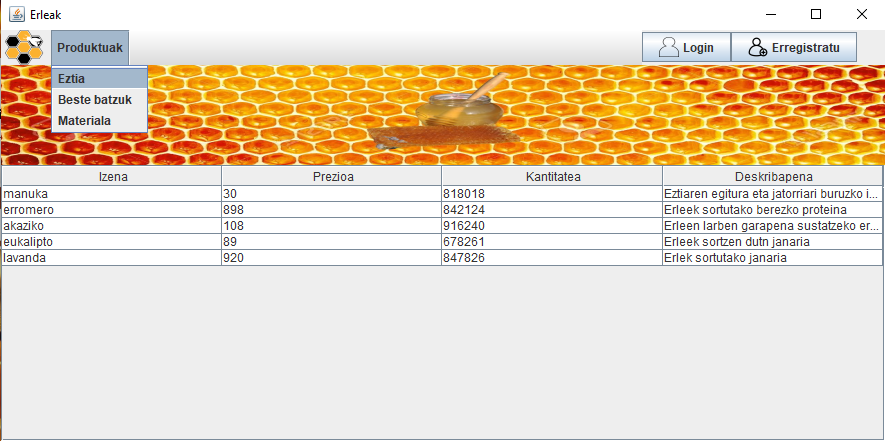
Bertan klik eginez gero, menu txiki bat irekiko da hiru botoirekin.



Irudia 46- Produktuak menua.

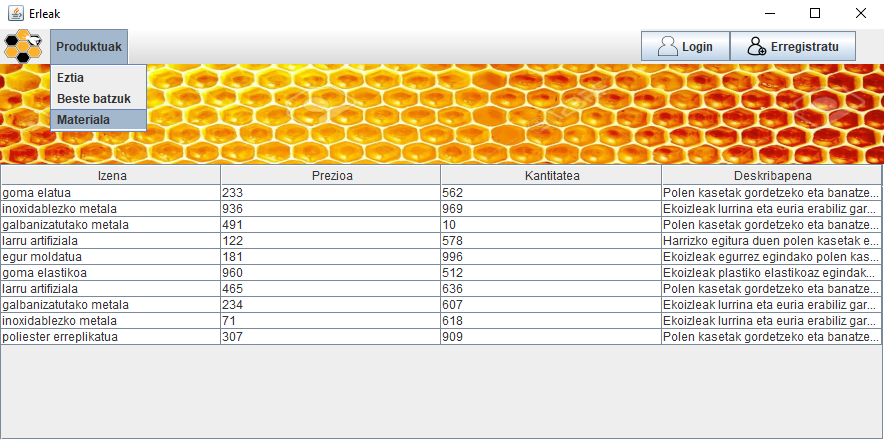
Botoi bakoitzari klik eginez gero, produktu horren taula bat pantailaratuko da bakoitzaren datuekin, honek datu basetik jasota.

Eztia klikatuz gero:



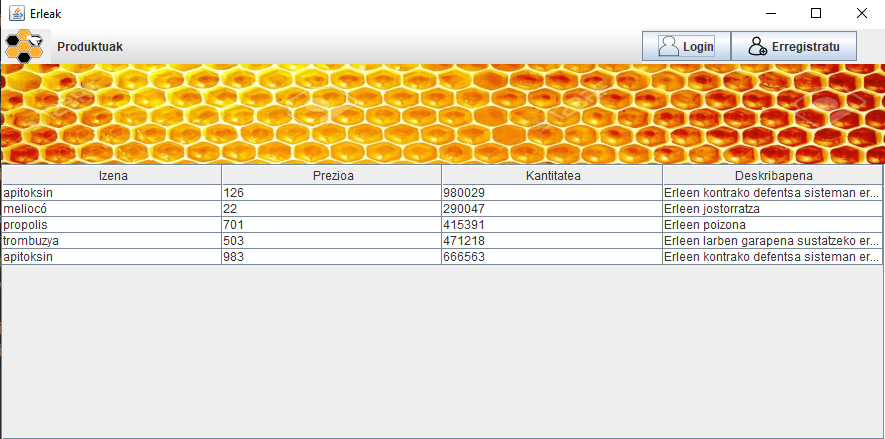
Irudia 47- Eztiak taula.

Materiala botoiaren gainean klik egiten bada:



Irudia 48- Materialak taula.

Azkenik beste batzuk hautatzen bada:



Irudia 49- Beste batzuk taula.