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

Erronka 2.-Sprinta

Aurkibidea:

[Triggerra 5](#_Toc164932469)

[Triggera pausuz pausu 6](#_Toc164932470)

[Log Triggera sortu 11](#_Toc164932471)

[Erabiltzaileak, taldeak eta karpetak kudeatu 17](#_Toc164932472)

[Taldeen sorrera 17](#_Toc164932473)

[Erabiltzaileen sorrera 18](#_Toc164932474)

[Karpeten sorrera 19](#_Toc164932475)

[Karpetei jabea jarri. 20](#_Toc164932476)

[Karpeten baimenak 22](#_Toc164932477)

[Partizioa 23](#_Toc164932478)

[Interfaze grafikoa 26](#_Toc164932479)

[Display nagusia. 26](#_Toc164932480)

[Erregistratu 27](#_Toc164932481)

[LogIn 28](#_Toc164932482)

[Produktuak 32](#_Toc164932483)

Irudien aurkibidea:

[Irudia 1-Erlauntz eta erle kopurua betetzen ez duen sozioa txertatu. 14](#_Toc164932412)

[Irudia 2-Erlauntz eta erle kantitate egokiak dituen sozioa txertatu. 14](#_Toc164932413)

[Irudia 3-Txertatutako sozioa sozioak taulan. 15](#_Toc164932414)

[Irudia 4-Sozioa historikoa txertatutako sozioa, log triggerraren bidez. 15](#_Toc164932415)

[Irudia 5-Erlauntz kantitatea betetzen ez duen sozioa txertatu. 15](#_Toc164932416)

[Irudia 6-Sozioa sozioak taulan txertatu eta triggeraren ondorioz erlauntz kopurua ezarri. 16](#_Toc164932417)

[Irudia 7-Sozioak\_historikoa taulan erregistroa log triggerraren ondorioz gehitu. 16](#_Toc164932418)

[Irudia 8-Sozioaren izena aldatu. 16](#_Toc164932419)

[Irudia 9-Izena aldatutako sozioaren izen berriari top gehitu. 16](#_Toc164932420)

[Irudia 10-Izena aldatutako sozioa parte den asoziazioen izenari top gehitu. 17](#_Toc164932421)

[Irudia 11-Taldeak sortu. 17](#_Toc164932422)

[Irudia 12- Makinako taldeak ikusi 18](#_Toc164932423)

[Irudia 13- Erabiltzaileak sortu eta taldeetan sartu 19](#_Toc164932424)

[Irudia 14- Erabiltzaileak ikusi 19](#_Toc164932425)

[Irudia 15- Karpetak sortu 20](#_Toc164932426)

[Irudia 16- Karpeten jabeak 20](#_Toc164932427)

[Irudia 17- Karpeten jabean 21](#_Toc164932428)

[Irudia 18- Karpeten baimenak 22](#_Toc164932429)

[Irudia 19- Diskoa gehitu 23](#_Toc164932430)

[Irudia 20- Diskoa ikusi 23](#_Toc164932431)

[Irudia 21- Diskoa ireki 24](#_Toc164932432)

[Irudia 22- Partizioa egin 24](#_Toc164932433)

[Irudia 23- Partizioa ikusi 25](#_Toc164932434)

[Irudia 24- Fitxategi sistema aldatu 25](#_Toc164932435)

[Irudia 25- Muntaketa puntua eta diskoaren muntaketa 25](#_Toc164932436)

[Irudia 26- Muntaketa puntu iraunkorra 26](#_Toc164932437)

[Irudia 27- Partizioa ikusi 26](#_Toc164932438)

[Irudia 28- Display nagusia. 27](#_Toc164932439)

[Irudia 29- Erregistratu botoia. 27](#_Toc164932440)

[Irudia 30- Erregistratu display-a. 27](#_Toc164932441)

[Irudia 31- Login botoia. 28](#_Toc164932442)

[Irudia 32- Login display. 28](#_Toc164932443)

[Irudia 33- Login errorea. 29](#_Toc164932444)

[Irudia 34- Login diplay 2. 29](#_Toc164932445)

[Irudia 35- Logeatu display. 29](#_Toc164932446)

[Irudia 36- Logeatuta display nagusia. 30](#_Toc164932447)

[Irudia 37- Profila. 30](#_Toc164932448)

[Irudia 38- Produktuak botoia. 32](#_Toc164932449)

[Irudia 39- Produktuak menua. 32](#_Toc164932450)

[Irudia 40- Eztiak taula. 33](#_Toc164932451)

[Irudia 41- Materialak taula. 33](#_Toc164932452)

[Irudia 42- Beste batzuk taula. 34](#_Toc164932453)

# 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

Lehenik, kontuan hartu behar da, bi “if” orokor daudela. 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. Bi **if** jarri behar dira, bat 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 “between” erabiltzea saihestu nahi da, orduan “and” eta “or” erabili dira.

|  |
| --- |
| 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; |

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. 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 “raise\_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” txertatu behar da. Hau ze\_ezti taulatik ikusiko da, bertan izena aldatutako sozioaren id-a konparatuko da eta “count” erabiliz aldi bakoitzean 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 da 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; |

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 ditu 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. Ondorengo log triggerra erabili da erreferentzia bezala eta gure datuetara moldatu da.

|  |
| --- |
| 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; |

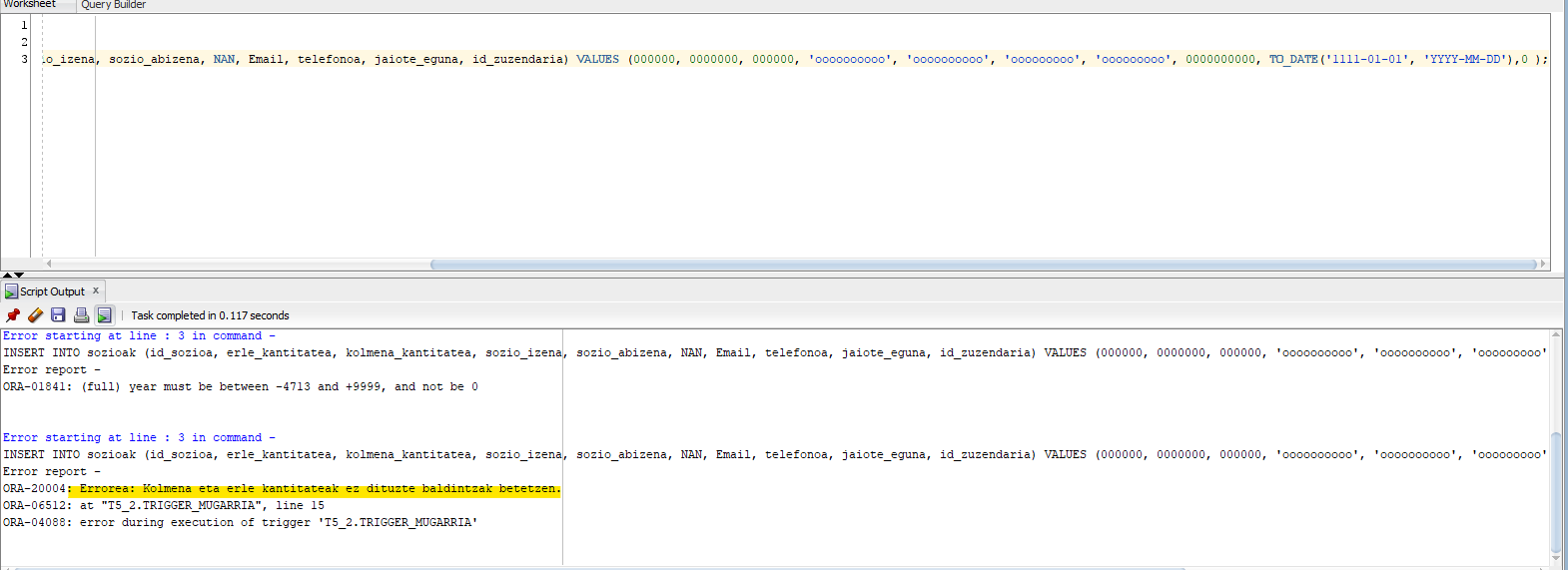
Lehenik, taularen izena aldatu, hemen bi taula izan behar ditugu kontuan: aldaketa egiten den taula eta aldaketa horren balio zaharrak eta berriak gordetzeko taula. Lehena sozioak taula izango da eta bertan gertatzen diren aldaketak sozioak\_historikoa taulan agertuko dira, bertan aldaketen datak jarriz.

|  |
| --- |
| 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. Aurrekoa 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.

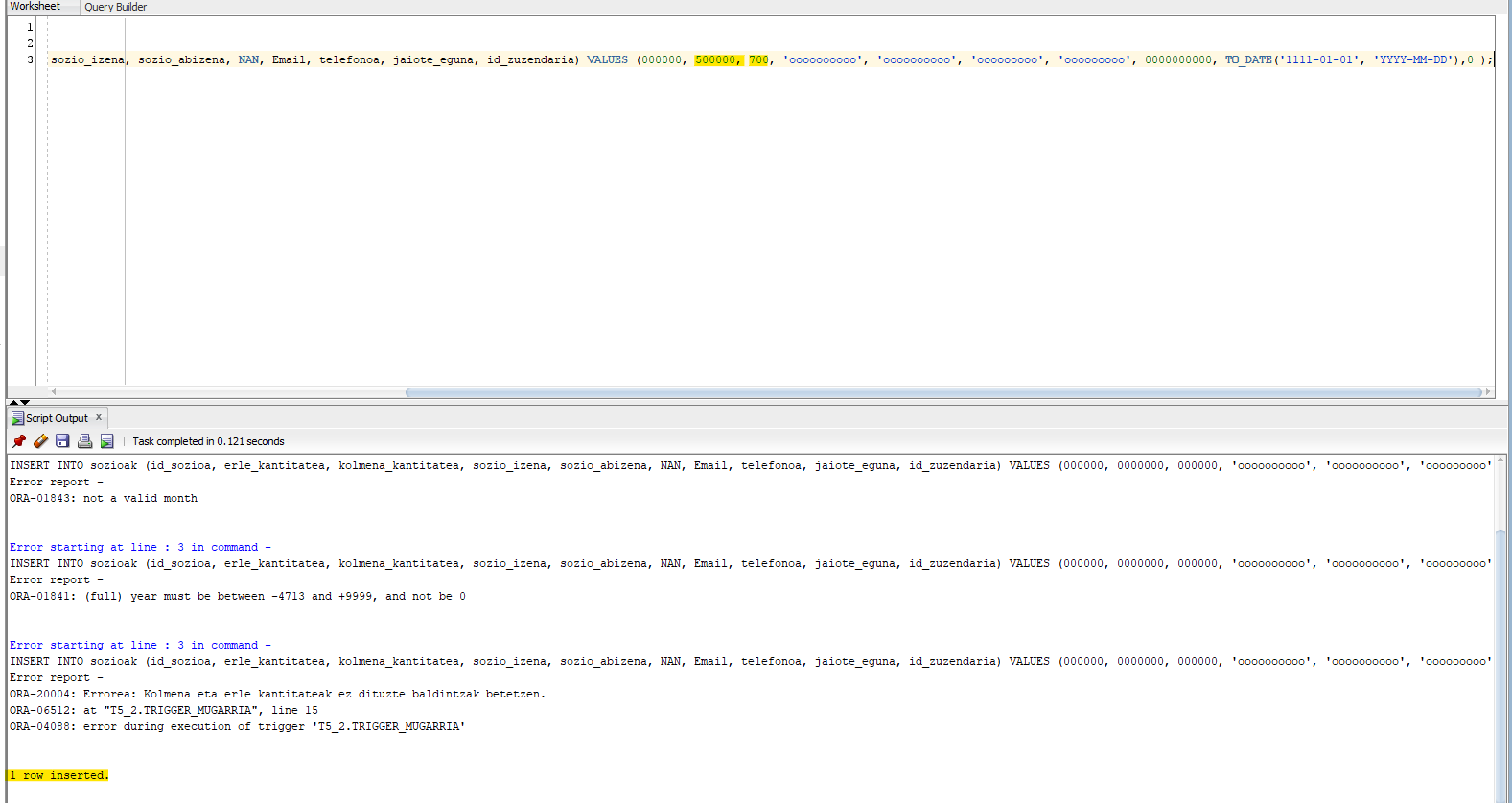
|  |
| --- |
| create or replace NONEDITIONABLE TRIGGER LOGTRIGGER\_ASOZIAZIOAK  AFTER INSERT OR UPDATE OR DELETE ON asoziazioak  FOR EACH ROW  DECLARE Now TIMESTAMP;  BEGIN  SELECT CURRENT\_TIMESTAMP INTO Now FROM Dual;  UPDATE asoziazioak\_historikoa  SET BukaeraData = Now  WHERE BukaeraData IS NULL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

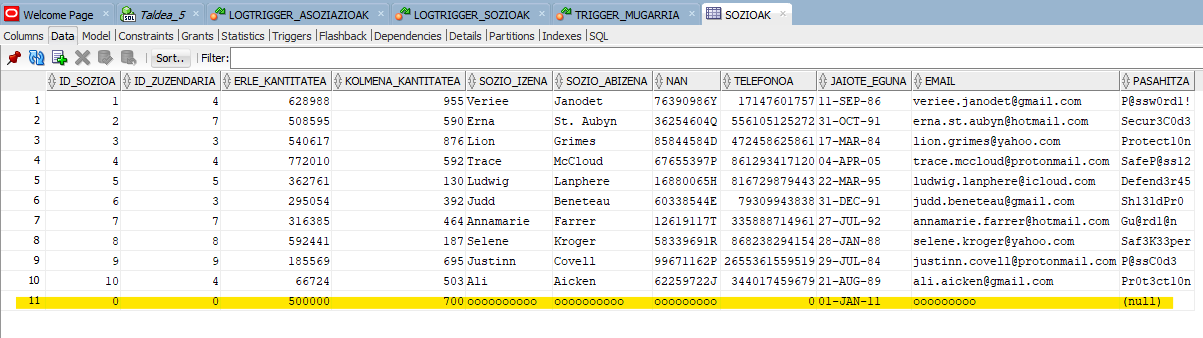


Irudia 1-Erlauntz eta erle kopurua betetzen ez duen sozioa txertatu.

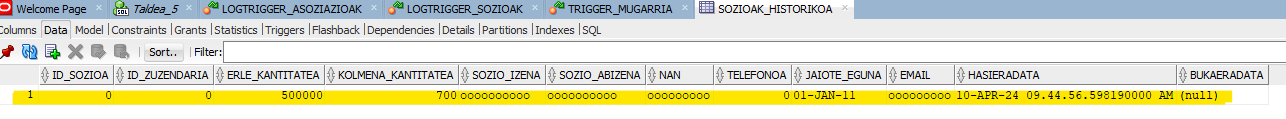
Erlauntz kantitatea eta erle kantitateak lehenengo baldintza betetzen badute baina bigarren ez, hau arazorik gabe txertatuko da, eta sozioak historikoan ere ongi txertatuko da.



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

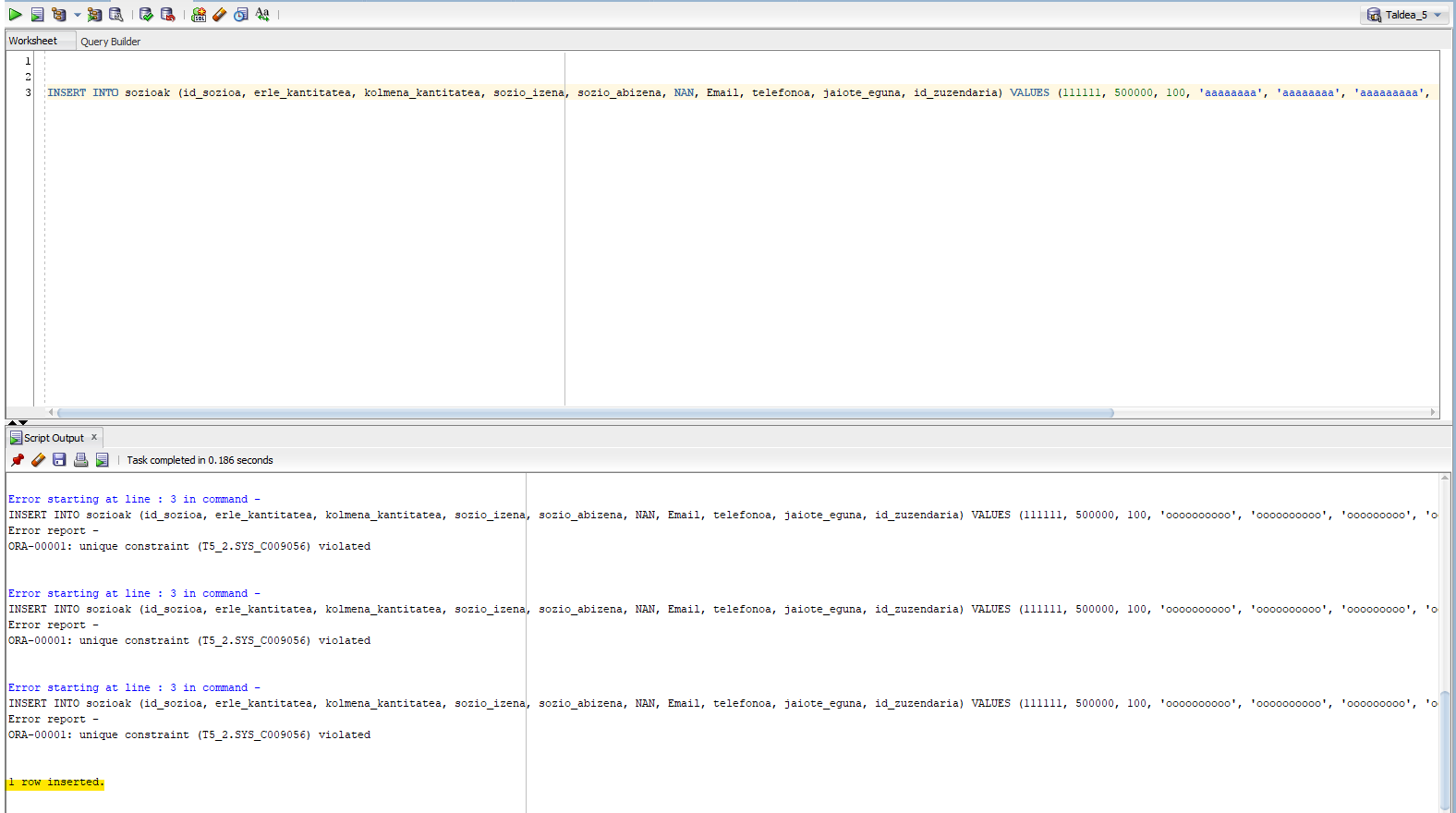


Irudia 3-Txertatutako sozioa sozioak taulan.



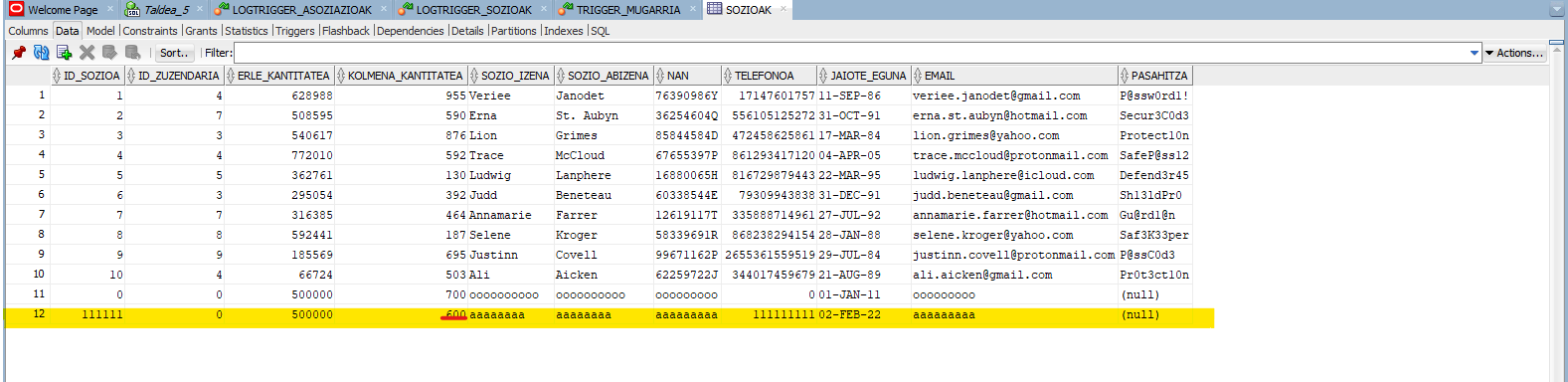
Irudia 4-Sozioa historikoan txertatutako sozioa, log triggerraren bidez.

Beste sozio bat sartu baina erlauntz kantitatea bete ez.

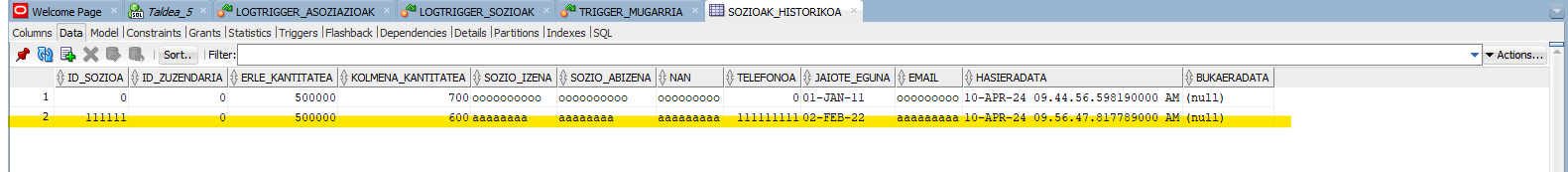


Irudia 5-Erlauntz kantitatea betetzen ez duen sozioa txertatu.

Triggeraren bidez erlauntz kantitatea 100 etik 600 era aldatu da.

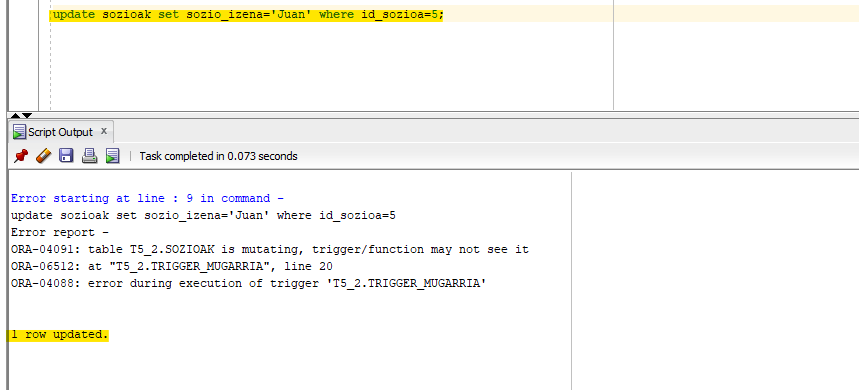


Irudia 6-Sozioa sozioak taulan txertatu eta triggeraren ondorioz erlauntz kopurua ezarri.



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

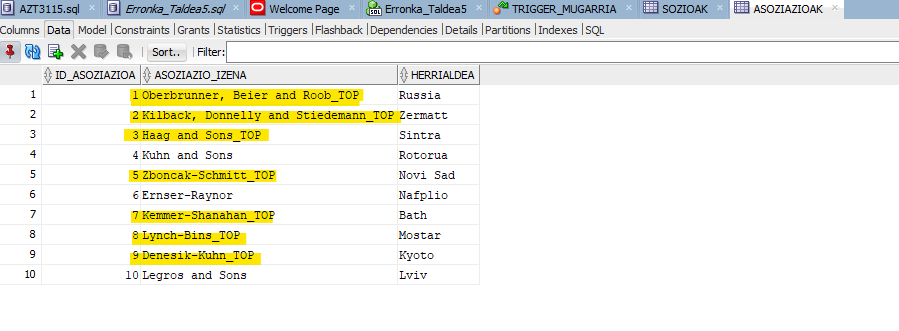
Triggerraren update ziurtatzeko sozio baten izena aldatu zaio. Honek bost produktu salduko ditu, berez izena aldatuko zaio baita bere asoziaziorena ere.



Irudia 8-Sozioaren izena aldatu.



Irudia 9-Izena aldatutako sozioaren izen berriari top gehitu.

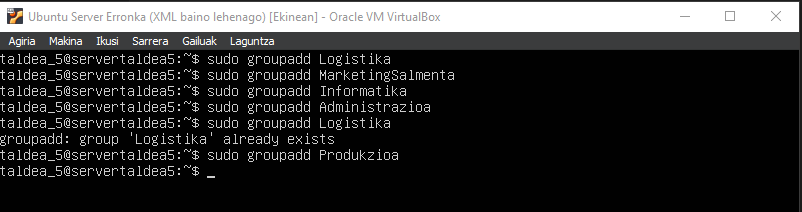


Irudia 10-Izena aldatutako sozioa parte den asoziazioen izenari top gehitu.

# Erabiltzaileak, taldeak eta karpetak kudeatu

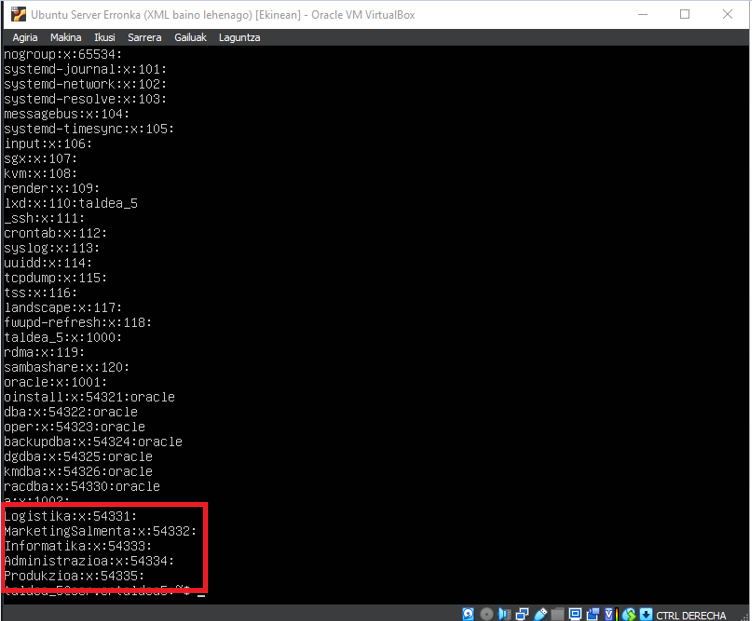
## Taldeen sorrera

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



Irudia 11-Taldeak sortu.

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



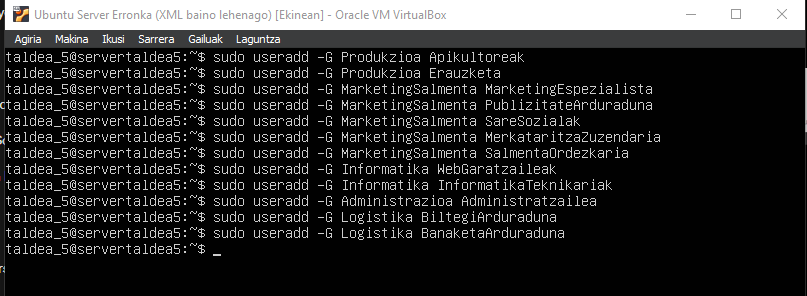
Irudia 12- Makinako taldeak ikusi

## Erabiltzaileen sorrera

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

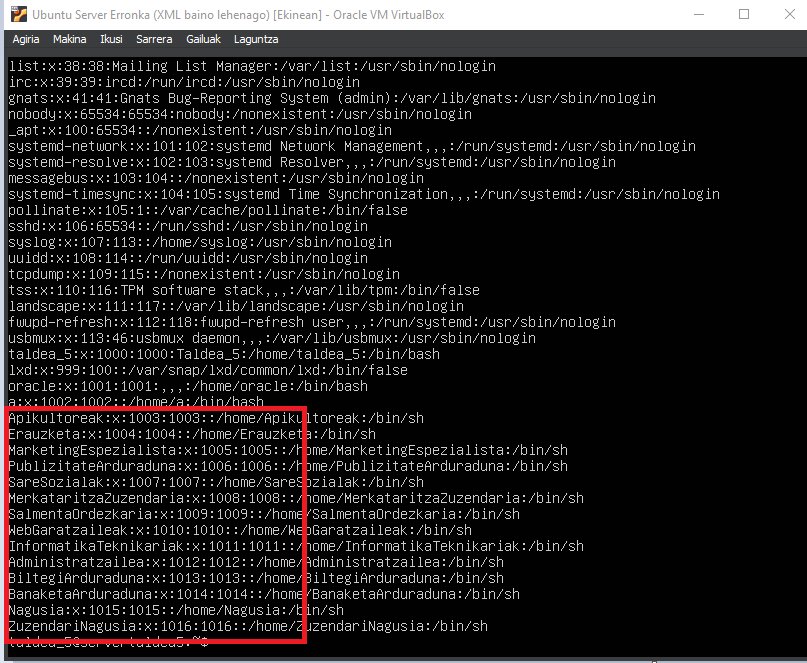
* Produkzioa taldea:
  + Apikultoreak
  + Erauzketa
* MarketingSalmenta taldea:
  + MarketingEzpezialista
  + SareSozialak
  + MerkataritzaZuzendaria
  + SalmentaOrdezkaria
* Irformatika taldea:
  + WebGaratzaileak
  + InformatikaGaratzaileak
* Logistika taldea:
  + BiltegiArduraduna
  + BanaketaArduraduna

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



Irudia 13- Erabiltzaileak sortu eta taldeetan sartu

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



Irudia 14- Erabiltzaileak ikusi

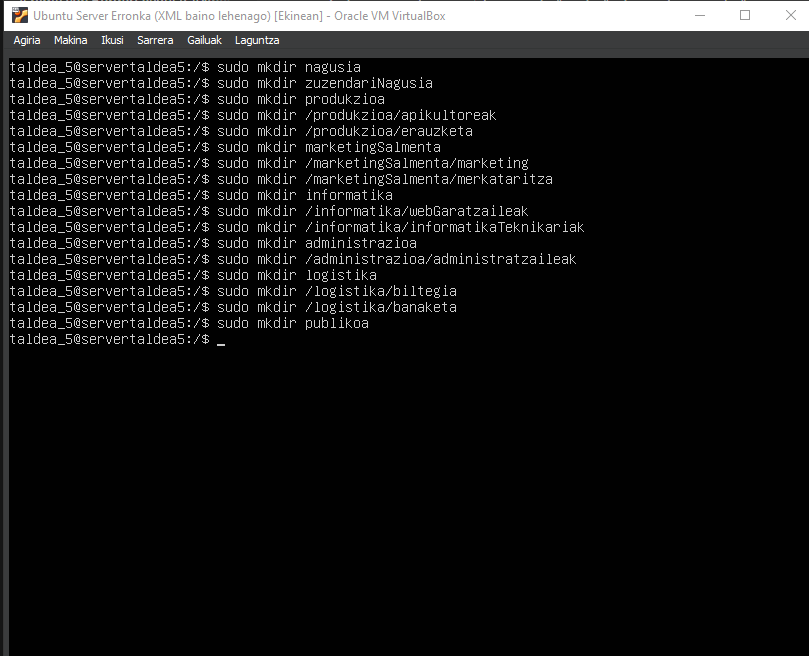
# Karpeten sorrera

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 dituzten karpetak izango dira.
* Karpeta publikoa: Enpresa osoan karpeta bat egongo da erabiltzaile guztientzat.

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

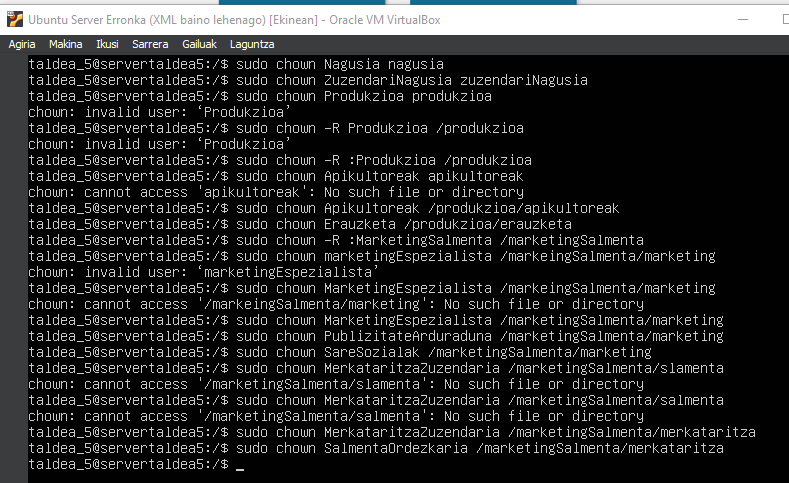


Irudia 15- Karpetak sortu

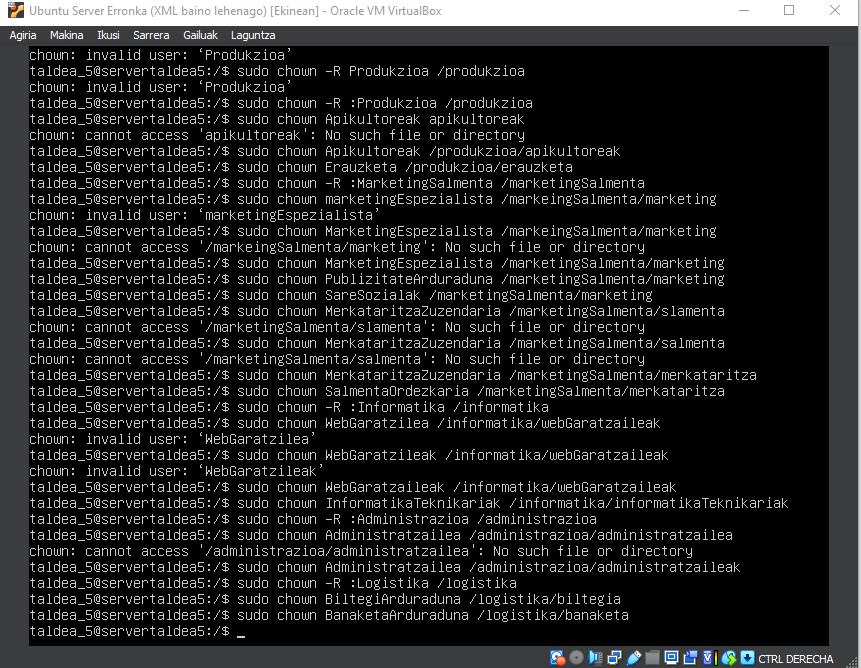
## Karpetei jabea ezarri.

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

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



Irudia 16- Karpeten jabeak



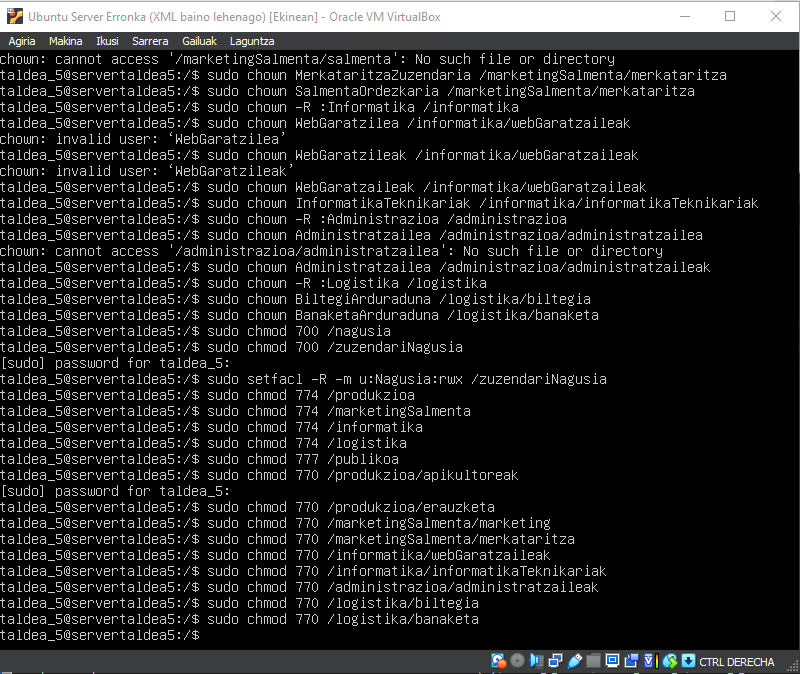
Irudia 17- Karpeten jabean

## Karpeten baimenak

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

Karpeta bakoitzaren baimenen arabera zenbaki ezberdina edukiko dute.

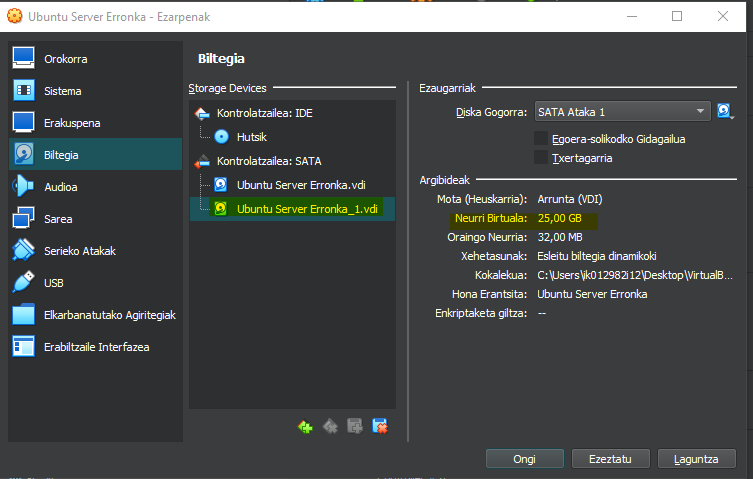
* 770 duten kasuan erabiltzaileak eta taldeak baimen denak edukiko dituzte, karpeta hauek erabiltzaileen karpetak izango dira.
* 774 taldeen karpetetan izango da, baimen denak erabiltzaile eta taldekoentzat eta beste edozeinek ikusteko aukera bakarrik izango dute .
* 777 duten karpetak, publikoak izango dira eta enpresa osoko erabiltzaile denek baimen guztiak edukiko dituzte.
* 700 baimenak dituzten karpetak egongo dira, hauek pribatuak direnez, jabeek edukiko dituzte karpetaren baimen guztiak.



Irudia 18- Karpeten baimenak

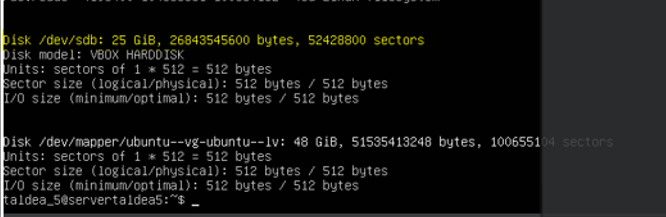
## Partizioa

Partizioa egiteko 25gb-ko diskoa gehitu zaio makina birtualari.



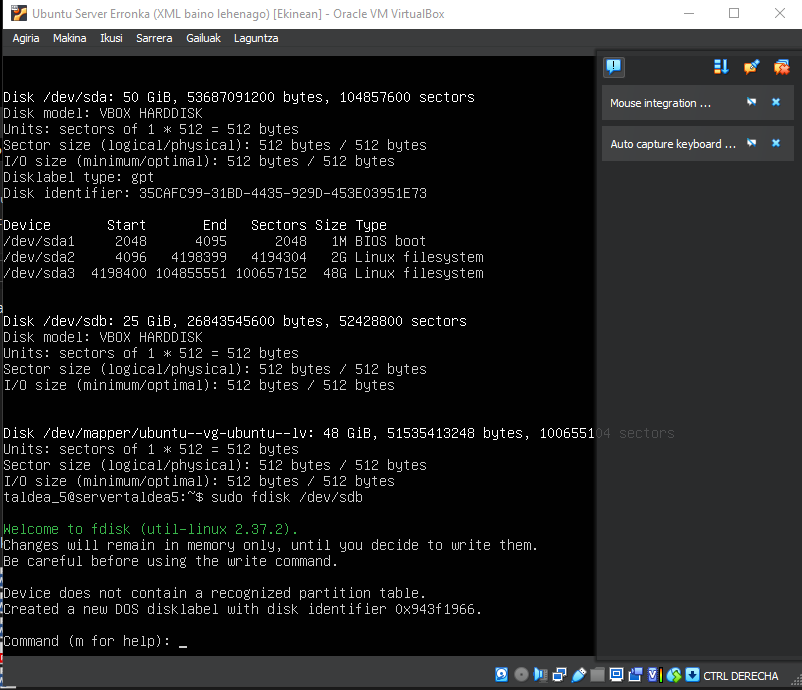
Irudia 19- Diskoa gehitu

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



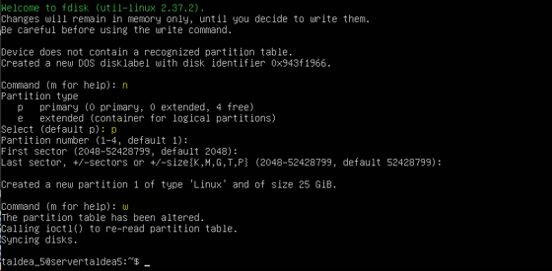
Irudia 20- Diskoa ikusi

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



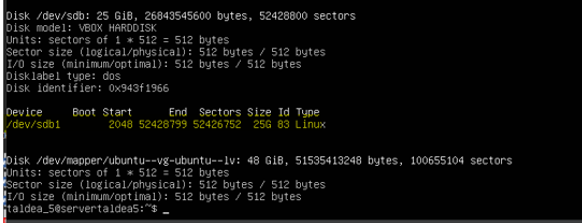
Irudia 21- Diskoa ireki

Diskoa irekita partizioa egiteari ekin zaio, horretarako **n** komandoa erabili da, ondoren “primary “motako partizioa hautatu da **p** bidez eta ondorengo aukera guztiak “default” utziz. Azkenik **w** komando exekutatu da partizioa gordetzeko.



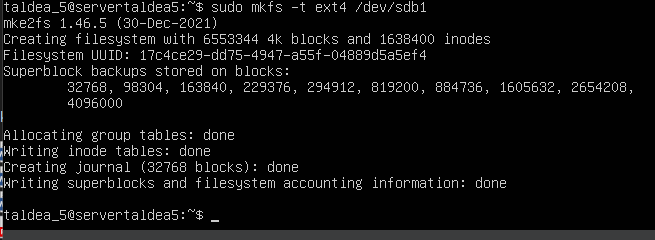
Irudia 22- Partizioa egiten

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



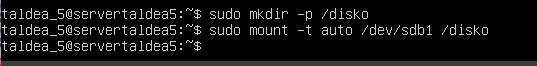
Irudia 23- Partizioa ikusi

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



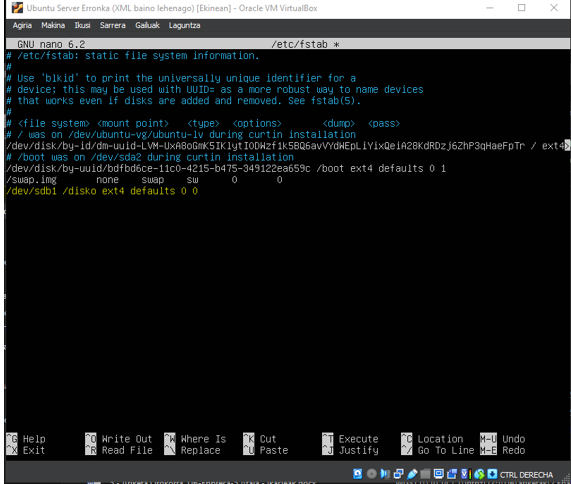
Irudia 24- Fitxategi sistema aldatu

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



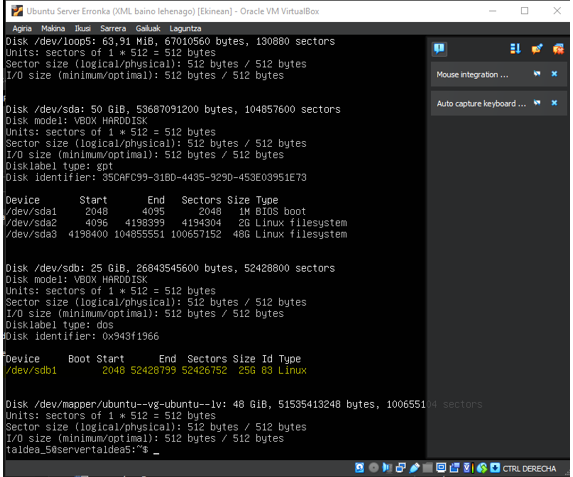
Irudia 25- Muntaketa puntua eta diskoaren muntaketa

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



Irudia 26- Muntaketa puntu iraunkorra

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

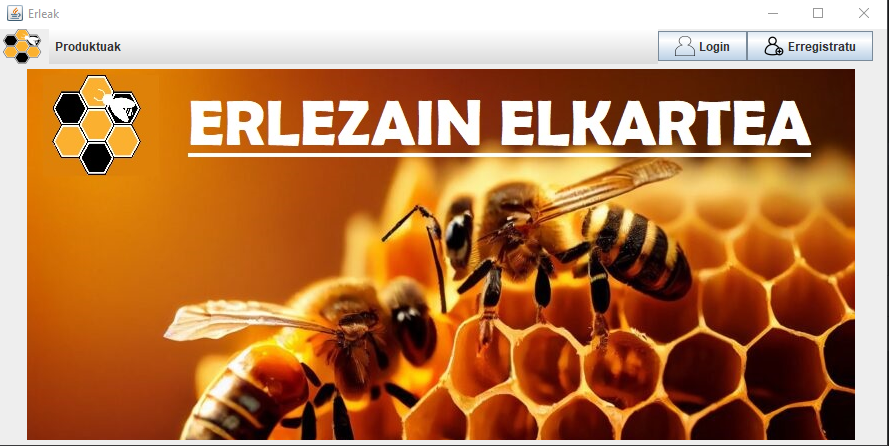


Irudia 27- Partizioa ikusi

# Interfaze grafikoa

## Display nagusia

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



Irudia 28- Display nagusia.

## Erregistratu

Alde batetik **erregistratu** botoia egongo da. Bertan klikatzerakoan beste display bat irekiko da bertan pertsonaren datuak sartzeko aukerarekin.

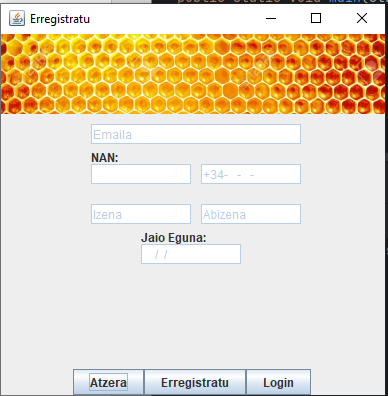


Irudia 29- Erregistratu botoia.

Hemen sozioaren datuak sartu ahalko dira eta erregistratu botoiari eman . Hau aurretik erregistratu ez den sozio batek egin beharko luke.

Honen azpian beste bi botoi daude. Alde batetik atzera botoia dago, honi klik egiterako momentuan haurreko display nagusira joango da.

Bestalde login botoia dago, erregistroa amaitutakoan logeatu ahal izateko.



Irudia 30- Erregistratu display-a.

## Login

Display-an dagoen beste botoia login botoia izango da.



Irudia 31- Login botoia.

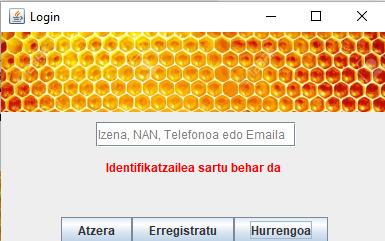
Botoiari klik eginez gero beste display bat irekiko da, non gure izena, nan-a, telefonoa edo email-a eskatuko den.

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



Irudia 32- Login display.

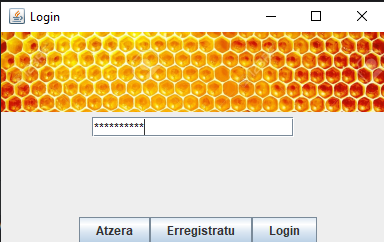
Ez bada identifikatzailerik sartzen ezingo da haurrera jarraitu.



Irudia 33- Login errorea.

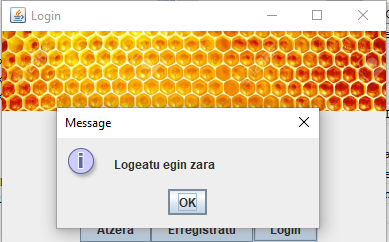
Pasahitz okerra sartuz gero login botoiak ez du ezer egingo beraz ezingo da logeatu.

Bestalde pasahitz zuzena sartuz gero sozioa logeatuko da.



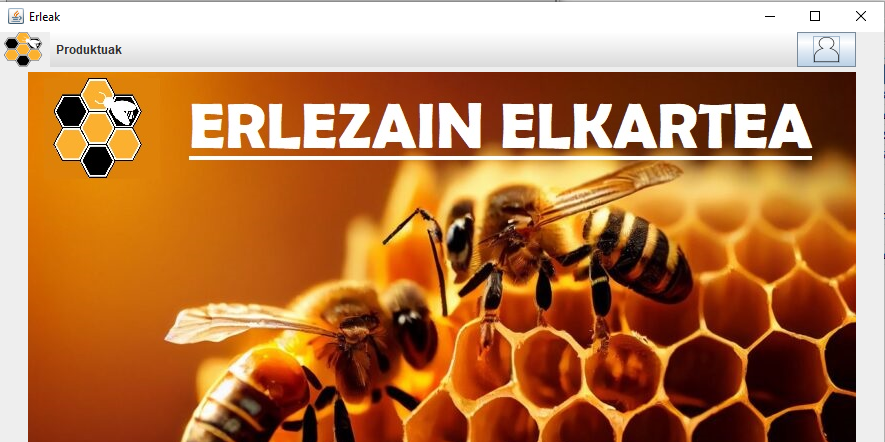
Irudia 34- Login diplay 2.

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



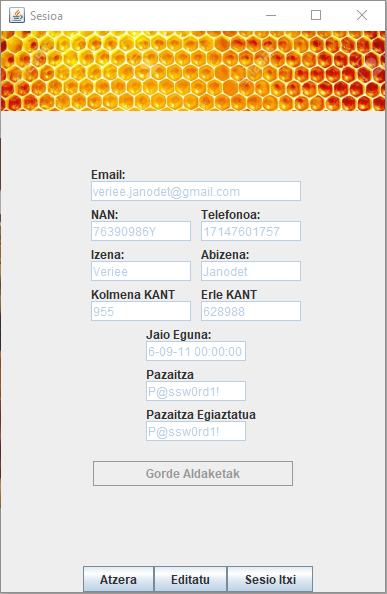
Irudia 35- Logeatu display.

Logeatu ondoren display nagusian profilaren botoia agertuko da.



Irudia 36- Logeatuta display nagusia.

Bertan klik egiten bada gure logeatutako sozioaren datuak pantailaratuko dira.

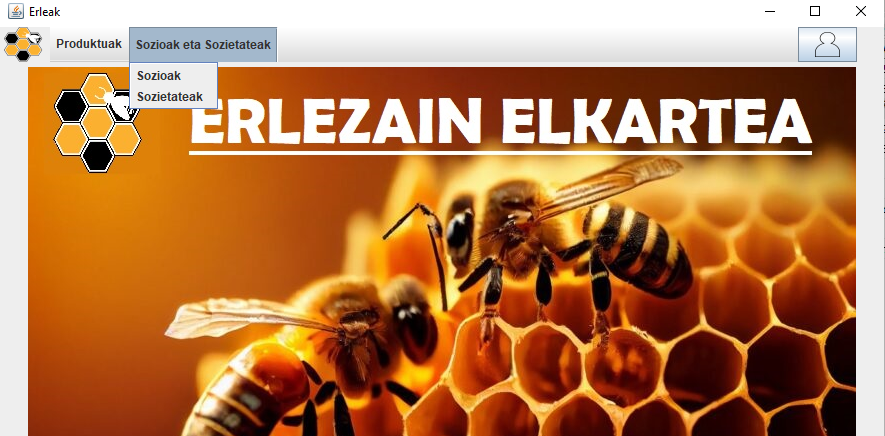


Irudia 37- Profila.

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.

Logeatutako sozioa zuzendaria bada, display nagusian beste menu botoi bat agertuko da sozioak eta sozietateak izenekoa.



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



Eta sozietateak botoian klik egiten bada bertan sozietate guztien informazioa bistaratuko da.



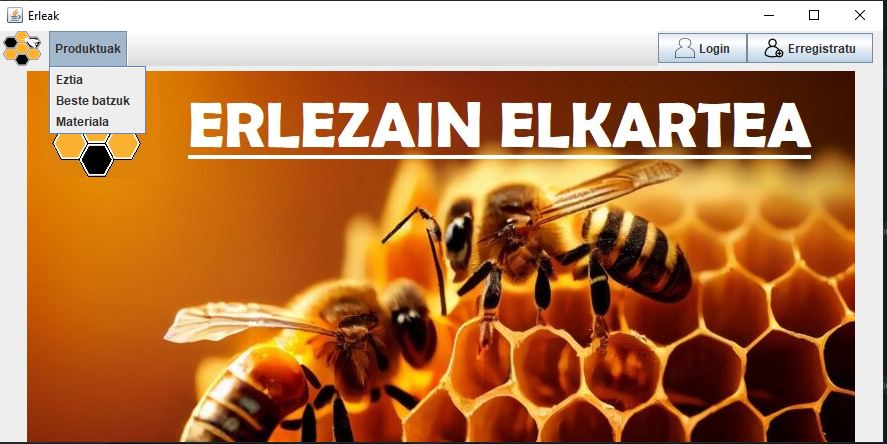
## Produktuak

Azkenik produktuak botoia ezango du.



Irudia 38- Produktuak botoia.

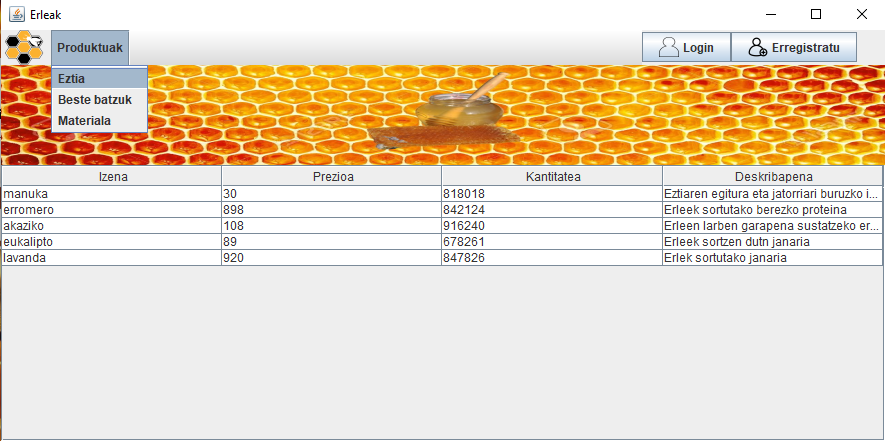
Bertan klik eginez gero, menu txiki bat irekiko da hiru botoirekin.



Irudia 39- Produktuak menua.

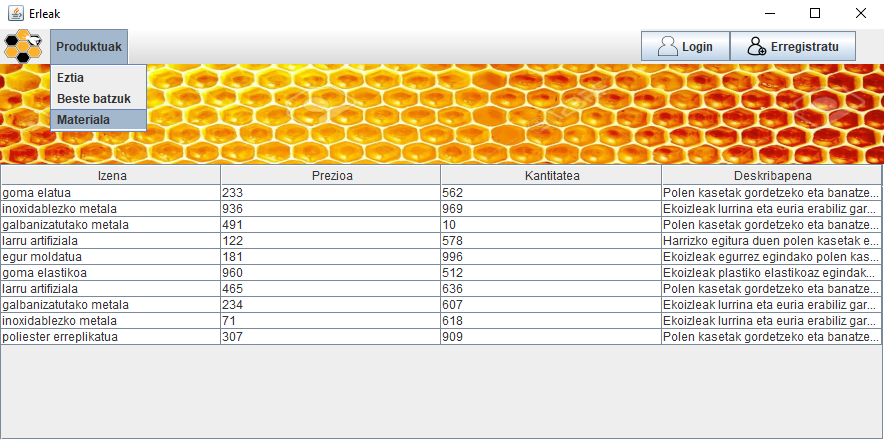
Botoi bakoitzari klik eginez gero, produktu horren taula bat pantailaratuko da bakoitzaren datuekin, honek datu basetik jasota.

Eztia klikatuz gero:



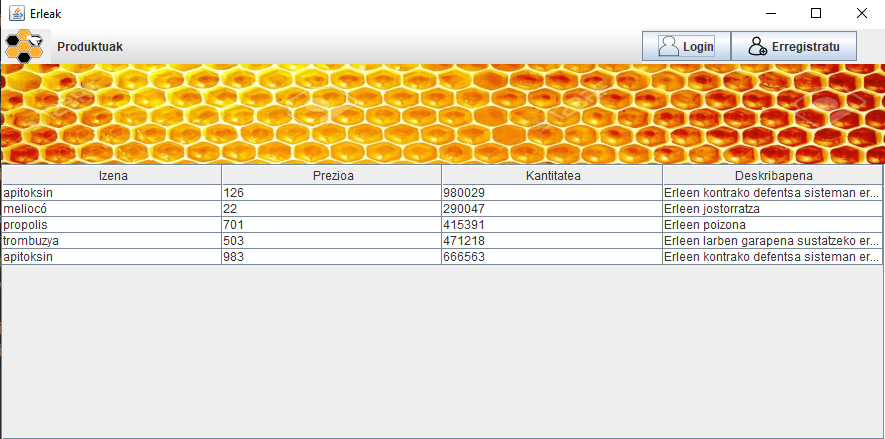
Irudia 40- Eztiak taula.

Materiala botoiaren gainean klik egiten bada:



Irudia 41- Materialak taula.

Azkenik beste batzuk hautatzen bada:



Irudia 42- Beste batzuk taula.

# XML Fitxategia

Eredu bezala erabiltzeko, parametro eta datu mota desberdin gehien dituen XML-a hautatu da.

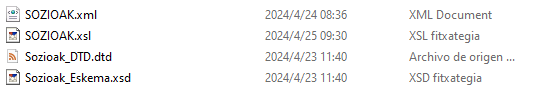


“Sozioak”, datuen egituraren erroa da, barnean sozioen erregistroak biltegiratuko dira.

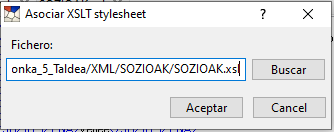
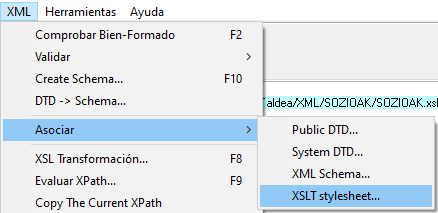
“Data\_record”-ek sozioen erregistro bakoitza definitzen du, barnean erregistroa definitzen duten elementuak izango ditu, erakutsi beharreko datuekin.

# XSL-a sortu eta XML-arekin lotu

Datu hauek html bidez webgunean bistaratu ahal izateko XSL fitxategi bat sortu da.



XSL fitxategia eraldatzen eta itxura ematen hasi aurretik, ondorengo irudietan ikusi daitekeen bezala XML fitxategia XSL-arekin lotu da.



Dena ondo egin bada, ondorengo lerroa gehituko da XML fitxategiaren kodean.



# XSL fitxategiaren edukia

XLS-ren bidez, XML dokumentuak formatuz aldatu daitezke, HTML, PDF edota beste XML dokumentu batzuetara eraldatuz.



Match atributua, fitxategia XML-ko elementu batekin lotzeko da, “Sozioak” XML fitxategiko erroa da, bere barneko datuak erakutsiko dira webgunean.

### HTML

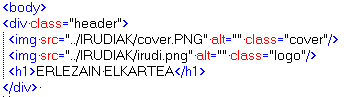
Hona ikusi daitekeen bezala, datuak tauletan erakutsiko dira html-egitura emanez.



* <xsl:for-each select=”DATA\_RECORD”>: Bukle bat erabiltzen da XML fitxategiko ‘DATA\_RECORD’ elementuetan iteratzeko, barnean duen html kodigoaren bidez, DATA\_RECORD bakoitzaren barneko erregistroak taula desberdinetan erakutsiko da.
* Xsl:value-of select: XSLT transformaziorako beharrezko elementua da. XML-tik erakutsi nahi diren balioak ateratzeko eta erakusteko erabiltzen bait da.

## KANPOKO ERREKURTSOAK

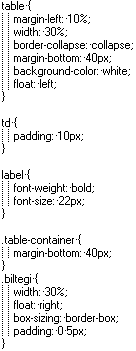
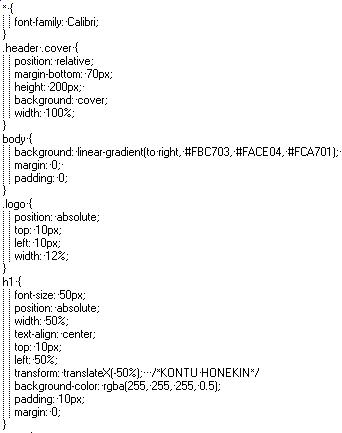
### IRUDIAK



Datuak bistaratzeko orduan, itxura egokia izan behar du, honetarako, lehenik, programan dagoen ‘banner’ berdina eta enpresaren logoaren irudia gehitu dira.

## CSS

CSS estilo batzuk gehitu dira html-aren edukia eta gehitutako irudiak modu egokian ikustarazteko, estilo nagusienak fondoko kolorea zehaztea, irudiak modu egokian kokatzea, taulak banakako lerroan agertu beharrean hirunaka ikustaraztea eta hauen fondoa zuria izatea izan dira. Hona hemen kodea:



## ITXURA GRAFIKOA



# DTD eta XML Schema

Fitxategi hauek XML-aren egitura eta baliozkotze arauak zehazten dituzten gida gisa jokatzen dute.

## DTD

XML fitxategi baten baimendutako datuak eta hauen egitura definitzen duen arau multzoa da.



XML dokumentoaren erroa (<Sozioak>) definitzen du eta barnean "data\_record" bat edo gehiago espero dituela zehazten da.



Erregistro bakoitzaren egitura eta barneko elementuen sekuentzia definitzen dira.



Elementu bakoitza (#PCDATA)-rekin definitua dago, beraz textu datuak bakarrik izan ditzakete.

## XML Schema

XML datuen egitura zehatzagoa adierazten du.

Eskema osatzean “Sozioak” elementu konplexua dela kontuan hartu behar da. Hori jakinda ‘sequence’ ireki eta barruan hamaika elementu egongo dira denak string motakoak, jaioteguna izan ezik, hau data bat izango denez ‘date’ motakoa izango da.

