```
USE IISSI_Friends;
-- da de alta a un usuario  con la fecha actual
DELIMITER //
create or replace procedure
pCreaUsuario(n varchar(50),fn DATE, e INTEGER, p INTEGER, g varchar(15),
em varchar(250),pa varchar(250), pe varchar(20),o varchar(20))
BEGIN

INSERT INTO
usuarios(nombre,fecha_nacimiento,estatura,genero,email,password,colorPelo,
colorOjo) values (n,fn,e,g,em,pa,pe,o);
END //
DELIMITER ;

-- elimina o da de baja a un usuario
DELIMITER //
create or replace PROCEDURE
pBorraUsuario(e VARCHAR(250))
BEGIN
DECLARE fp INTEGER;
DECLARE f INTEGER ;
DECLARE v INTEGER;
DECLARE u INTEGER;
DECLARE s INTEGER;
DECLARE fa DATE;
SET fa = CURDATE();
SET fp = (SELECT COUNT(*) FROM fichasDePreferencias  WHERE email = e);
SET f = ( SELECT COUNT(*) FROM fotos  WHERE email = e);
SET v = (SELECT COUNT(*) FROM vinculos_amistades   WHERE email1 = e OR
email2 = e);
SET u = (SELECT COUNT(*) FROM ubicaciones WHERE email = e);
SET s = fp+f+v+u+s;
if(s>0)then
UPDATE  usuarios SET fechaBaja = fa,estaActivo= false WHERE email = e;
ELSE  DELETE FROM usuarios WHERE email = e;
END if ;
END //
DELIMITER ;


-- añadir ubicacion
DELIMITER //
create or replace PROCEDURE
pInsertaUbicacion(e VARCHAR(250),cp INT(5),m VARCHAR(20), p VARCHAR(20))
BEGIN
INSERT INTO ubicaciones(email,codigoPostal,municipio,provincia) VALUES
(e,cp,m,p);
END //
DELIMITER ;


-- modificar ubicación
DELIMITER //
create or replace PROCEDURE
```

```
pModificarUbicacion(uID INT(11),cp INT(5),m VARCHAR(20), p VARCHAR(20))
BEGIN
UPDATE ubicaciones SET codigoPostal =cp ,municipio= m,provincia = p WHERE
ubicacionID = uID;
END //
DELIMITER ;

-- eliminar ubicacion
DELIMITER //
create or replace PROCEDURE
pEliminaUbicacion(uID INTEGER(11))
BEGIN
DELETE FROM ubicaciones where ubicacionID = uID;
END //
DELIMITER ;
-- crear ficha
DELIMITER //
create or replace PROCEDURE
pAñadeFicha(e VARCHAR(250),eMax INTEGER, eMin INTEGER ,cp INT(5), m
VARCHAR(20), prov VARCHAR(20),
estMax INTEGER, estMin INTEGER, pmax INTEGER, pMin INTEGER,colP
VARCHAR(20),colO VARCHAR(20), g VARCHAR(20))
BEGIN
INSERT INTO
fichasDePreferencias(edad_maxima,edad_minima,codigoPostal,municipio,provin
cia,estatura_maxima,
estatura_minima,peso_maximo, peso_minimo,email ,colorPelo
,colorOjo,genero) VALUES
(eMAx,eMin,cp,m,prov,estMax,estMin,pMAx,pMin,e,colP,colO,g);
END //
DELIMITER ;


-- elimina ficha
DELIMITER //
create or replace PROCEDURE
pEliminaFicha(fID INT(11))
BEGIN
DELETE FROM fichasDePreferencias where fichaID = fID;
END //
DELIMITER ;



-- inserta la foto de un usuario con la fecha actual
DELIMITER //
create or replace procedure
pInsertaFoto(n varchar(50),u varchar(250),e varchar(250), d varchar(500))
BEGIN

INSERT INTO fotos(nombre,url,email,descripcion) values (n,u,e,d);
END //
DELIMITER ;
```

```
-- modifica el nombre,la url y la descripcion
DELIMITER //
create or replace procedure
pModificaFoto(fID INT(11), n varchar(50),u varchar(250),d varchar(500))
BEGIN
UPDATE fotos set nombre= n, url = u, descripcion = d where fotoID = fID;
END //
DELIMITER ;

-- elimina una foto

DELIMITER //
create or replace procedure
pEliminarFoto(fID int(11))
BEGIN
DELETE from fotos where fotoID= fID;
END//
DELIMITER ;
 -- elimina todas las fotos del usuario
DELIMITER //
CREATE  OR REPLACE procedure
pEliminaFotosUsuario(e varchar(250))
BEGIN
delete from fotos where email = e;
END //
DELIMITER ;

-- PROCEDURE para añadir aficion a usuario
DELIMITER //
create OR REPLACE PROCEDURE
pAnadirAficionUsuario(a INT(11),e varchar(250))
BEGIN
INSERT INTO aficiones_usuarios(aficionID,email) VALUES (a,e);
END //
DELIMITER ;

-- procedure para modificar aficcion a usuario
DELIMITER //
create OR REPLACE PROCEDURE
pModificaAficionUsuario(AuID INT(11),AFid INT(11))
BEGIN
UPDATE aficiones_usuarios P  SET aficionID = AFid WHERE aficionesUsuarioID
= AuID;
END //
DELIMITER ;

-- procedure para eliminar aficion a usuario
DELIMITER //
create OR REPLACE PROCEDURE
pEliminaAficionUsuario(AuID INT(11))
BEGIN
DELETE FROM aficiones_usuarios WHERE aficionesUsuarioID = AuID;
END //
DELIMITER ;
```

```
-- procedure para eliminar todas las aficiones de un usuario
DELIMITER //
create OR REPLACE PROCEDURE
pEliminaTodasAficionesUsuario(e VARCHAR(250))
BEGIN
DELETE FROM aficiones_usuarios WHERE email = e;
END //


-- Procedure para añadir aficion a ficha
DELIMITER //
create OR REPLACE PROCEDURE
pAnadeAficionFicha(fID INT(11),aID INT(11))
BEGIN
INSERT INTO aficiones_preferidas(aficionID,fichaID) VALUES(fID,aID);

END //
DELIMITER ;
-- procedure para modificar aficcion a ficha
DELIMITER //
create OR REPLACE PROCEDURE
pModificaAficionFicha(AFID INT(11),AID INT(11))
BEGIN
UPDATE aficiones_preferidas  SET aficionID = AFid WHERE
aficionesPreferidasID = AID;
END //
DELIMITER ;

-- procedure para eliminar aficion a ficha
DELIMITER //
create OR REPLACE PROCEDURE
pEliminaAficionFicha(AuID INT(11))
BEGIN
DELETE FROM aficiones_preferidas WHERE aficionesPreferidasID = AuID;
END //
DELIMITER ;

-- procedure para eliminar todas aficion a ficha
DELIMITER //
CREATE OR REPLACE PROCEDURE
pEliminaTodasAficionesFicha(fID INT(11))
BEGIN
DELETE FROM aficiones_preferidas WHERE fichaID = fID;
END //
DELIMITER ;


-- procedure para añiadir aficion
DELIMITER //
create OR REPLACE PROCEDURE

 pCreaAficion(a VARCHAR(20))
 BEGIN
 INSERT INTO aficiones(aficion) VALUES (a);
```

```
 END //
DELIMITER ;

-- procedure para modificar aficion

DELIMITER //
create OR REPLACE PROCEDURE
pModificaAficion(id INT(11), a VARCHAR(20))
BEGIN
UPDATE aficiones SET aficion = a WHERE( aficionID = id);

 END //
DELIMITER ;

-- procedure para eliminar aficion
DELIMITER //
create OR REPLACE PROCEDURE
pEliminaAficion(id INT(11))
BEGIN
DELETE FROM aficiones WHERE aficionID = id;
END//
DELIMITER ;

-- crear convesrascion
DELIMITER //
CREATE OR REPLACE PROCEDURE
pIniciaConversacion(vId INT(11))
BEGIN
declare f date;
declare b Boolean;
set f = CURDATE();
set b = true;
INSERT INTO conversaciones(vinculoID,fecha_inicio,new_message) VALUES(
vId,f,b);
END //
DELIMITER ;
-- finalizar conversación

DELIMITER //
create OR REPLACE PROCEDURE
pFinalizaConversacion(cID INT(11))
BEGIN
DECLARE n DATE;
SET n = CURDATE();
UPDATE conversaciones SET fecha_fin = n WHERE (conversacionID = cID);
END //
DELIMITER ;

-- crear mensaje
DELIMITER //
create OR REPLACE PROCEDURE
pAnadirMensaje(cID INT(11), t VARCHAR(500))

BEGIN
```

```
DECLARE i DATETIME;
SET i = CURRENT_TIMESTAMP();
INSERT INTO mensajes(conversacionID,instante,texto) VALUES (cID,i,t);
END //
DELIMITER ;
-- crear solicitud
DELIMITER //
CREATE OR REPLACE PROCEDURE
pSolicitaVinculo(e1 VARCHAR(250), e2 VARCHAR(250))
BEGIN
DECLARE d DATE ;
SET d = CURDATE();
INSERT INTO vinculos_amistades(emai1,email2, fecha_solicitud,
fecha_aceptación,fecha_revocacion_solicitud,fecha_revocacion_aceptación,
VinculoActivo)VALUES (
e1,e2,d,NULL,NULL,NULL,FALSE);
END //
DELIMITER ;


-- aceptar solicitud
DELIMITER //
CREATE OR REPLACE PROCEDURE
pAceptarSolicitud(vID INT(11))
BEGIN
DECLARE i DATE;
SET i = CURDATE();
UPDATE vinculos_amistades SET fecha_solicitud= i,VinculoActivo = TRUE
WHERE vinculoID = vID;
END //
DELIMITER ;


-- revocar solicitud
DELIMITER //
CREATE OR REPLACE PROCEDURE
pRevocarSolicitud(vID INT(11))
BEGIN
DECLARE i DATE;
SET i = CURDATE();
UPDATE vinculos_amistades SET fecha_revocacion_solicitud= i, VinculoActivo
= FALSE  WHERE vinculoID = vID;
END //
DELIMITER ;

-- revocar aceptacion
DELIMITER //
CREATE OR REPLACE PROCEDURE
pRevocarAceptacion(vID INT(11))
BEGIN
DECLARE i DATE;
SET i = CURDATE();
UPDATE vinculos_amistades SET fecha_revocacion_aceptacion= i,
VinculoActivo = FALSE  WHERE vinculoID = vID;
```

```
END //
DELIMITER ;

-- ver solicitudes pendientes de un usuario (email)


CREATE OR REPLACE VIEW SolPendientes
 AS SELECT  * from vinculos_amistades  WHERE fecha_aceptación = NULL;



DELIMITER //
CREATE OR REPLACE PROCEDURE
pVerSolicitudesPendientes(e VARCHAR(250))
BEGIN
SELECT * FROM SolPendientes  WHERE  email2 = e ;
END //
DELIMITER ;



--  ver vinculos activos de un usuario
/* creo que la estructura es parecida al anterior aunque convendria vsacar
ademas una
vista con los vinculos que esten activos exclusivamente y utilizarla.
Ademas en el where seria where email1 = email or email2 = email
*/


CREATE OR REPLACE VIEW VincActivos AS
SELECT * FROM vinculos_amistades
      WHERE  vinculos_amistades.fecha_aceptación IS not NULL AND
vinculos_amistades.fecha_revocacion_solicitud IS not NULL OR
vinculos_amistades.fecha_revocacion_aceptación IS not NULL ;

DELIMITER //
create or replace procedure
pVerVinculosActivos( e varchar(250))
BEGIN
Select * FROM VincActivos where email1 = e or email2 = e;
END //
DELIMITER ;

-- explorar vinculos historicos
/*
la vista que usariamos sería una con los vinulos historicos
*/
CREATE OR REPLACE VIEW vinculosHistoricos AS
SELECT * FROM vinculos_amistades WHERE fecha_revocacion_solicitud = NULL
OR fecha_revocacion_aceptación = NULL;
DELIMITER //
create or replace procedure
pVerVinculosHistoricos(e varchar(250))
BEGIN
```

```
Select * from  vinculosHistoricos where emai1 = e or email2 = e;
END //
DELIMITER ;


--  algorimo de emarejamiento
/* parecida a las anteriores pero mas compleja
hay que crear una funcion que nos diga si tiene ficha de preferenias y
segun el resultado  hacer la bifurcacion con el if
segun cada caso
*/
```