CREATE OR replace PROCEDURE insertusuario (pidusuario IN INT,

pnombre IN VARCHAR,

papellido1 IN VARCHAR,

papellido2 IN VARCHAR,

pcontraseña IN VARCHAR)

AS

BEGIN

INSERT INTO usuario

(

idusuario,

nombre,

apellido1,

apellido2,

contrasena

)

VALUES

(

pidusuario,

pnombre,

papellido1,

papellido2,

pcontraseña

);

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END insertusuario;

---------------------------------------------------------------------------------------------------------------------------

create or replace PROCEDURE insertestudiante (pidusuario IN INT,

pnombre IN VARCHAR,

papellido1 IN VARCHAR,

papellido2 IN VARCHAR,

pcontraseña IN VARCHAR,

pfoto IN VARCHAR,

pidsede IN INT,

ptelefono IN VARCHAR,

pcorreo IN VARCHAR,

pdescripcion IN VARCHAR,

ptelefonosecundario IN VARCHAR,

pcorreosecundario IN VARCHAR)

AS

BEGIN

insertusuario(pidusuario,pnombre,papellido1,papellido2,pcontraseña);

INSERT INTO estudiante

(

descripcion,

fotografia,

fechaingreso,

cantidadtecolones,

correo,

telefono,

idestado,

idusuario,

idsede,

cantidadcolones

)

VALUES

(

pdescripcion,

pfoto,

sysdate,

0,

pcorreo,

ptelefono,

1,

pidusuario,

pidsede,

0

);

inserttelefono(ptelefonosecundario,pidusuario);

insertcorreo(pcorreosecundario,pidusuario);

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END insertestudiante;

---------------------------------------------------------------------------------------------------------------------------

create or replace FUNCTION loginestudiante (pidusuario IN INT,

pcontrasena IN VARCHAR)

RETURN VARCHAR

IS

retorno VARCHAR(50);

BEGIN

BEGIN

SELECT estado

INTO retorno

FROM estado, estudiante, usuario

WHERE estudiante.idusuario=pidusuario

AND estudiante.idusuario=usuario.idusuario

AND usuario.contrasena=pcontrasena

AND estudiante.idestado=estado.idestado;

END;

RETURN retorno;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END loginestudiante;

------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE OR replace FUNCTION loginadministrador (pidusuario IN INT,

pcontrasena IN VARCHAR)

RETURN INT

IS

retorno INT;

BEGIN

BEGIN

SELECT departamento.iduniversidad

INTO retorno

FROM administrador, usuario, departamento

WHERE administrador.idusuario=pidusuario

AND administrador.idusuario=usuario.idusuario

AND usuario.contrasena=pcontrasena

AND administrador.iddepartamento=departamento.iddepartamento;

END;

RETURN retorno;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END loginadministrador;

--------------------------------------------------------------------------------------------------------------------------------------------------------------

create or replace FUNCTION loginadministradoracopio (pidusuario IN INT,

pcontrasena IN VARCHAR)

RETURN INT

IS

retorno INT;

BEGIN

BEGIN

SELECT centroacopio.idsede

INTO retorno

FROM centroacopio, administradoracopio, usuario

WHERE administradoracopio.idusuario=pidusuario

AND administradoracopio.idusuario=usuario.idusuario

AND usuario.contrasena=pcontrasena AND administradoracopio.idcentroacopio = centroacopio.idcentroacopio;

END;

RETURN retorno;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END loginadministradoracopio;

----------------------------------------------------------------------------------------------------------------------------------------------------------------

----------------NUEVAS --------------------------------------------------------------

create or replace PROCEDURE insertadministradoracopio (pidusuario IN INT,

pnombre IN VARCHAR,

papellido1 IN VARCHAR,

papellido2 IN VARCHAR,

pcontraseña IN VARCHAR,

pidcentroacopio IN INT)

AS

BEGIN

insertusuario(pidusuario,pnombre,papellido1,papellido2,pcontraseña);

INSERT INTO administradoracopio

(

idusuario,

idcentroacopio

)

VALUES

(

pidusuario,

pidcentroacopio

);

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END insertadministradoracopio;

--------------------------------------------------------------------------------------------------------------

create or replace PROCEDURE insertsede (pnombre IN VARCHAR,

pubicacion IN VARCHAR,

piduniversidad IN INT)

AS

BEGIN

INSERT INTO sede

(

nombre,

ubicacion,

iduniversidad

)

VALUES

(

pnombre,

pubicacion,

piduniversidad

);

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END insertsede;

---------------------------------------------------------------------------------------------------------------

create or replace PROCEDURE insertcentroacopio (pidsede IN INT,

pnombre IN VARCHAR)

AS

BEGIN

INSERT INTO centroacopio

(

idsede,

nombre

)

VALUES

(

pidsede,

pnombre

);

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END insertcentroacopio;

---------------------------------------------------------------------------------------------------------------

create or replace PROCEDURE insertuniversidad (pnombre IN VARCHAR)

AS

BEGIN

INSERT INTO universidad

(

nombre,

meta

)

VALUES

(

pnombre,

20

);

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END insertuniversidad;

--------------------------------------------------------------------------------------------------------------

create or replace PROCEDURE insertadministrador (pidusuario IN INT,

pnombre IN VARCHAR,

papellido1 IN VARCHAR,

papellido2 IN VARCHAR,

pcontraseña IN VARCHAR,

piddepartamento IN INT)

AS

BEGIN

insertusuario(pidusuario,pnombre,papellido1,papellido2,pcontraseña);

INSERT INTO administrador

(

idusuario,

iddepartamento

)

VALUES

(

pidusuario,

piddepartamento

);

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END insertadministrador;

----------------------------------------------------------------------------------------------------------

CREATE OR replace PROCEDURE inserttipotweet (ptipotweet IN INT)

AS

BEGIN

INSERT INTO tipotweet

(

tipotweet

)

VALUES

(

ptipotweet

);

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END inserttipotweet;

--------------------------------------------------------------------------------------------------------------------

CREATE OR replace PROCEDURE inserttweet (pcuerpo IN VARCHAR,

ptipotweet IN INT)

AS

BEGIN

INSERT INTO tweet

(

cuerpo,

idtipotweet

)

VALUES

(

pcuerpo,

ptipotweet

);

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END inserttweet;

-----------------------------------------------------------------------------------------------------------------------

CREATE OR replace PROCEDURE insertcuentatwitter (pconsumerkey IN VARCHAR,

pconsumersecret IN VARCHAR,

paccesstoken IN VARCHAR,

paccesstokensecret IN VARCHAR,

piduniversidad IN INT)

AS

BEGIN

INSERT INTO cuentatwitter

(

consumerkey,

consumersecret,

accesstoken,

accesstokensecret,

iduniversidad

)

VALUES

(

pconsumerkey,

pconsumersecret,

paccesstoken,

paccesstokensecret,

piduniversidad

);

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END insertcuentatwitter;

--------------------------------------------------------------------------------------------------------------------------

CREATE OR replace PROCEDURE insertdepartamento (pdepartamento IN VARCHAR)

AS

BEGIN

INSERT INTO departamento

(

departamento

)

VALUES

(

pdepartamento

);

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END insertdepartamento;

---------------------------------------------------------------------------------------------------------------------------

CREATE OR replace PROCEDURE inserttipocambio (ptipocambio IN FLOAT)

AS

BEGIN

INSERT INTO tipocambio

(

tipocambio

)

VALUES

(

ptipocambio

);

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END inserttipocambio;

---------------------------------------------------------------------------------------------------------------------------

CREATE OR replace PROCEDURE insertestado (pestado IN VARCHAR)

AS

BEGIN

INSERT INTO estado

(

estado

)

VALUES

(

pestado

);

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END insertestado;

---------------------------------------------------------------------------------------------------------------------------

CREATE OR replace PROCEDURE insertmaterial (pnombre IN VARCHAR,

pvalor IN FLOAT)

AS

BEGIN

INSERT INTO material

(

nombre,

valor

)

VALUES

(

pnombre,

pvalor

);

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END insertmaterial;

---------------------------------------------------------------------------------------------------------------------------

CREATE OR replace PROCEDURE insertestudiantematerial (pidusuario IN INT,  
 pidsede IN INT,  
 pidmaterial IN INT,  
 pcantidad IN FLOAT)  
AS  
BEGIN  
updatetecolones(pidusuario, pcantidad, pidmaterial);  
 INSERT INTO estudiantematerial  
 (  
 idusuario,  
 idsede,  
 idmaterial,  
 cantidad,  
 fecharegistro   
 )  
 VALUES  
 (  
 pidusuario,  
 pidsede,  
 pidmaterial,  
 pcantidad,  
 sysdate  
 );  
   
 COMMIT;  
EXCEPTION  
WHEN OTHERS THEN  
 ROLLBACK;  
END insertestudiantematerial;

---------------------------------------------------------------------------------------------------------------------------

CREATE OR replace PROCEDURE updatetecolones (pidusuario IN INT,

pcantidad IN FLOAT,

pidmaterial IN INT)

AS

valor NUMBER;

BEGIN

SELECT material.valor INTO valor FROM material WHERE material.idmaterial=pidmaterial;

UPDATE estudiante

SET estudiante.cantidadtecolones = estudiante.cantidadtecolones+pcantidad\*valor

WHERE estudiante.idusuario=pidusuario;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END updatetecolones;

---------------------------------------------------------------------------------------------------------------------------

CREATE OR replace PROCEDURE returncentroacopio (pidsede IN INT,

c\_centroacopio OUT sys\_refcursor)

IS

BEGIN

OPEN c\_centroacopio FOR

SELECT idcentroacopio,nombre

FROM centroacopio

WHERE idsede=pidsede;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END returncentroacopio;

-------------------------------------------------------------------------------------------------------------------------

create or replace PROCEDURE rebajartecolones (pidusuario IN INT,  
 pidusuario2 IN INT,  
 pcantidad IN FLOAT)  
IS  
 D\_no\_tecolones EXCEPTION;  
 cantidad NUMBER;  
 c NUMBER;  
BEGIN  
  
 SELECT estudiante.cantidadtecolones INTO cantidad FROM estudiante WHERE estudiante.idusuario=pidusuario;  
 SELECT count(estudiante.idusuario) into c from usuario,estudiante where usuario.idusuario=pidusuario2 and estudiante.idusuario=usuario.idusuario;  
IF cantidad<pcantidad or c=0 THEN  
 RAISE D\_no\_tecolones;  
 ELSE  
 UPDATE estudiante  
 SET estudiante.cantidadtecolones = estudiante.cantidadtecolones-pcantidad  
 WHERE estudiante.idusuario=pidusuario AND estudiante.cantidadtecolones>=pcantidad;  
 END IF;  
 COMMIT;  
END rebajartecolones;​

---------------------------------------------------------------------------------------------------------------------------

CREATE OR replace PROCEDURE aumentartecolones (pidusuario IN INT,

pcantidad IN FLOAT)

AS

BEGIN

UPDATE estudiante

SET estudiante.cantidadtecolones = estudiante.cantidadtecolones+pcantidad

WHERE estudiante.idusuario=pidusuario;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END aumentartecolones;

---------------------------------------------------------------------------------------------------------------------------

CREATE OR replace PROCEDURE returnsede(piduniversidad IN INT,

c\_sede OUT sys\_refcursor)

IS

BEGIN

OPEN c\_sede FOR

SELECT idsede,nombre

FROM sede

WHERE iduniversidad=piduniversidad;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END returnsede;

---------------------------------------------------------------------------------------------------------------------------

CREATE OR replace PROCEDURE returnuniversidad(c\_u OUT sys\_refcursor)

IS

BEGIN

OPEN c\_u FOR

SELECT \*

FROM universidad COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END returnuniversidad;

---------------------------------------------------------------------------------------------------------------------------

create or replace FUNCTION returntoneladasanuales (piduniversidad IN INT,

pyear IN INT)

RETURN FLOAT

IS

retorno FLOAT;

BEGIN

BEGIN

SELECT SUM(estudiantematerial.cantidad)/1000

INTO retorno

FROM estudiantematerial, sede

WHERE estudiantematerial.idsede=sede.idsede AND sede.iduniversidad=piduniversidad AND extract(year from estudiantematerial.fecharegistro) = pyear;

END;

RETURN retorno;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END returntoneladasanuales;

---------------------------------------------------------------------------------------------------------------------------

//esta NO

create or replace FUNCTION returntoneladassede (pidsede IN INT)

RETURN FLOAT

IS

retorno FLOAT;

BEGIN

BEGIN

SELECT SUM(estudiantematerial.cantidad)/1000

INTO retorno

FROM estudiantematerial

WHERE estudiantematerial.idsede=pidsede;

END;

RETURN retorno;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END returntoneladassede;

---------------------------------------------------------------------------------------------------------------------------

create or replace FUNCTION returnestudiantesmensuales (piduniversidad IN INT, pmonth IN INT)

RETURN FLOAT

IS

retorno FLOAT;

BEGIN

BEGIN

SELECT COUNT(DISTINCT(idusuario))

INTO retorno

FROM estudiantematerial

WHERE extract(month from estudiantematerial.fecharegistro) = pmonth;

END;

RETURN retorno;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END returnestudiantesmensuales;

---------------------------------------------------------------------------------------------------------------------------

CREATE OR replace PROCEDURE returntop(piduniversidad IN INT,

c\_top OUT sys\_refcursor)

IS

BEGIN

OPEN c\_top FOR

SELECT \*

FROM( SELECT idusuario, nombre, apellido1, apellido2, sum(cantidad) as suma

FROM (SELECT estudiantematerial.idusuario, usuario.nombre, usuario.apellido1, usuario.apellido2, estudiantematerial.cantidad

FROM estudiantematerial, usuario, sede where estudiantematerial.idusuario = usuario.idusuario and estudiantematerial.idsede=sede.idsede and sede.iduniversidad=piduniversidad)

group by (idusuario, nombre, apellido1, apellido2) order by sum(cantidad) desc) WHERE ROWNUM<=10;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END returntop;

----------------------------------------------

---------------------------------------------------------------------------------------------------------------------------

CREATE OR replace PROCEDURE returndesgloseanual(piduniversidad IN INT,

pyear IN INT,

c\_mensual OUT sys\_refcursor)

IS

BEGIN

OPEN c\_mensual FOR

SELECT nombre, sum(cantidad)/1000

FROM (SELECT material.nombre, estudiantematerial.cantidad

FROM estudiantematerial, material, sede where extract(year from estudiantematerial.fecharegistro) = pyear and

estudiantematerial.idmaterial = material.idmaterial and estudiantematerial.idsede=sede.idsede and sede.iduniversidad=piduniversidad)

group by (nombre) order by sum(cantidad) desc;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END returndesgloseanual;

---------------------------------------------------------------------------------------------------------------------------

create or replace PROCEDURE returndesglose(pidsede IN INT,

pmes IN INT,

pano IN INT,

c\_mensual OUT sys\_refcursor)

IS

BEGIN

OPEN c\_mensual FOR

SELECT nombre, sum(cantidad)/1000

FROM (SELECT material.nombre, estudiantematerial.cantidad

FROM estudiantematerial, material, sede

where extract(year from estudiantematerial.fecharegistro) = pano and extract(month from estudiantematerial.fecharegistro)=pmes and

estudiantematerial.idmaterial = material.idmaterial and estudiantematerial.idsede=pidsede)

group by (nombre) order by sum(cantidad) desc;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END returndesglose;

---------------------------------------------------------------------------------------------------------------------------

create or replace PROCEDURE returntoneladassede(piduniversidad IN INT,

c\_mensual OUT sys\_refcursor)

IS

BEGIN

OPEN c\_mensual FOR

SELECT nombre, sum(cantidad)/1000

FROM (SELECT sede.nombre, estudiantematerial.cantidad

FROM estudiantematerial, sede

where extract(year from estudiantematerial.fecharegistro) = extract(year from sysdate) and extract(month from estudiantematerial.fecharegistro)=extract(month from sysdate)

and estudiantematerial.idsede=sede.idsede and sede.iduniversidad=piduniversidad)

group by (nombre) order by sum(cantidad) desc;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END returntoneladassede;​

---------------------------------------------------------------------------------------------------------------------------

---------------------------------------------------------------------------------------------------------------------------

import java.io.IOException;

import twitter4j.Status;

import twitter4j.Twitter;

import twitter4j.TwitterException;

import twitter4j.TwitterFactory;

import twitter4j.conf.ConfigurationBuilder;

Funcion Java para twittear

public void twittear(String ck, String cs, String at, String ats, String tweet) throws IOException, TwitterException{

Twitter twitter;

ConfigurationBuilder cb = new ConfigurationBuilder();

cb.setDebugEnabled(true)

.setOAuthConsumerKey(ck)

.setOAuthConsumerSecret(cs)

.setOAuthAccessToken(at)

.setOAuthAccessTokenSecret(ats);

twitter = new TwitterFactory(cb.build()).getInstance();

Status tweetEscrito = twitter.updateStatus(tweet);

}

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

create or replace PROCEDURE returntweetmaterial(piduniversidad IN INT,

pidusuario IN INT,

pidmaterial IN INT,

c\_mensual OUT sys\_refcursor)

IS

BEGIN

OPEN c\_mensual FOR

SELECT material.nombre, usuario.nombre as usuario, tweet.cuerpo, cuentatwitter.consumerkey, cuentatwitter.consumersecret, cuentatwitter.accesstoken, cuentatwitter.accesstokensecret

FROM material, usuario, tweet, cuentatwitter, tipotweet

where tipotweet.idtipotweet=2 and tweet.idtipotweet=tipotweet.idtipotweet and cuentatwitter.iduniversidad=piduniversidad and usuario.idusuario=pidusuario and material.idmaterial=pidmaterial;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END returntweetmaterial;

---------------------------------------------------------------------------------------------------------------------------

create or replace PROCEDURE returntweetcompartir(piduniversidad IN INT,

pidusuario IN INT,

c\_mensual OUT sys\_refcursor)

IS

BEGIN

OPEN c\_mensual FOR

SELECT usuario.nombre, tweet.cuerpo, cuentatwitter.consumerkey, cuentatwitter.consumersecret, cuentatwitter.accesstoken, cuentatwitter.accesstokensecret

FROM tweet, cuentatwitter, tipotweet, usuario

where usuario.idusuario=pidusuario and tipotweet.idtipotweet=2 and tweet.idtipotweet=tipotweet.idtipotweet and cuentatwitter.iduniversidad=piduniversidad;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END returntweetcompartir;

---------------------------------------------------------------------------------------------------------------------------

CREATE OR replace PROCEDURE insertpromocionsimple

(

piduniversidad IN INT,

pidmaterial IN INT,

pcantidad IN NUMBER,

pbono IN NUMBER,

pfechainicio IN VARCHAR,

pfechafinal IN VARCHAR)

AS

c INT;

BEGIN

DELETE from promocion where (idmaterial1=pidmaterial or idmaterial2=pidmaterial) and (TO\_DATE(pfechainicio,'dd/mm/yyyy') BETWEEN fechainicio and fechafinal or TO\_DATE(pfechafinal,'dd/mm/yyyy') BETWEEN fechainicio and fechafinal);

insert into promocion

(

iduniversidad,

idmaterial1,

cantidad1,

bono,

fechainicio,

fechafinal,

tipo)

VALUES(

piduniversidad,

pidmaterial,

pcantidad,

pbono,

TO\_DATE(pfechainicio,'dd/mm/yyyy'),

TO\_DATE(pfechafinal,'dd/mm/yyyy'),

'simple');

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END insertpromocionsimple;

---------------------------------------------------------------------------------------------------------------------------

CREATE OR replace PROCEDURE insertpromociondoble

(

piduniversidad IN INT,

pidmaterial1 IN INT,

pidmaterial2 IN INT,

pcantidad1 IN NUMBER,

pcantidad2 IN NUMBER,

pbono IN NUMBER,

pfechainicio IN VARCHAR,

pfechafinal IN VARCHAR)

AS

BEGIN

DELETE from promocion where ((idmaterial1=pidmaterial1 or idmaterial2=pidmaterial1) or (idmaterial1=pidmaterial2 or idmaterial2=pidmaterial2)) and (TO\_DATE(pfechainicio,'dd/mm/yyyy') BETWEEN fechainicio and fechafinal or TO\_DATE(pfechafinal,'dd/mm/yyyy') BETWEEN fechainicio and fechafinal);

insert into promocion

(

iduniversidad,

idmaterial1,

idmaterial2,

cantidad1,

cantidad2,

bono,

fechainicio,

fechafinal,

tipo)

VALUES(

piduniversidad,

pidmaterial1,

pidmaterial2,

pcantidad1,

pcantidad2,

pbono,

TO\_DATE(pfechainicio,'dd/mm/yyyy'),

TO\_DATE(pfechafinal,'dd/mm/yyyy'),

'doble');

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END insertpromociondoble;

---------------------------------------------------------------------------------------------------------------------------

CREATE OR replace PROCEDURE updatebono (pidusuario IN INT,

pcantidad IN FLOAT)

AS

BEGIN

UPDATE estudiante

SET estudiante.cantidadtecolones = estudiante.cantidadtecolones+pcantidad

WHERE estudiante.idusuario=pidusuario;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END updatebono;

---------------------------------------------------------------------------------------------------------------------------

CREATE OR replace PROCEDURE returndepartamento(piduniversidad IN INT,

c\_departamento OUT sys\_refcursor)

IS

BEGIN

OPEN c\_departamento FOR

SELECT iddepartamento,departamento

FROM departamento

WHERE iduniversidad=piduniversidad;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END returndepartamento;

---------------------------------------------------------------------------------------------------------------------------

CREATE OR replace PROCEDURE updateIntentologin (pidusuario IN INT)

AS

BEGIN

UPDATE usuario

SET usuario.intento = usuario.intento+1

WHERE usuario.idusuario=pidusuario;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END updateIntentologin;

---------------------------------------------------------------------------------------------------------------------------

CREATE OR replace PROCEDURE updateIntentocerrarsesion (pidusuario IN INT)

AS

BEGIN

UPDATE usuario

SET usuario.intento = 0

WHERE usuario.idusuario=pidusuario;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END updateIntentocerrarsesion;

---------------------------------------------------------------------------------------------------------------------------

create or replace FUNCTION returnintento (pidusuario IN INT)

RETURN INT

IS

retorno INT;

BEGIN

BEGIN

SELECT intento

INTO retorno

FROM usuario

WHERE idusuario=pidusuario;

END;

RETURN retorno;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END returnintento;

---------------------------------------------------------------------------------------------------------------------------

create or replace PROCEDURE insertbono (pidusuario IN INT,

pidpromocion IN INT)

AS

BEGIN

INSERT INTO bono

(

idusuario,

idpromocion

)

VALUES

(

pidusuario,

pidpromocion

);

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END insertbono;

---------------------------------------------------------------------------------------------------------------------------

CREATE OR replace PROCEDURE registrarmaterial (pidusuario IN INT,

pidsede IN INT,

pidmaterial IN INT,

pcantidad IN NUMBER)

AS

c INT;

c2 INT;

promo INT;

tp varchar2(10);

material1 INT;

material2 INT;

cant1 NUMBER;

cant2 NUMBER;

micant1 NUMBER;

micant2 NUMBER;

bonus NUMBER;

BEGIN

insertestudiantematerial(pidusuario,pidsede,pidmaterial,pcantidad);

select count(idpromocion) into c from promocion where sysdate between promocion.fechainicio and promocion.fechafinal and (idmaterial1=pidmaterial or idmaterial2=pidmaterial);

select count(idbono) into c2 from bono where idpromocion=promo and idusuario=pidusuario;

select idpromocion into promo from promocion where sysdate between promocion.fechainicio and promocion.fechafinal and (idmaterial1=pidmaterial or idmaterial2=pidmaterial);

select idmaterial1 into material1 from promocion where sysdate between promocion.fechainicio and promocion.fechafinal and (idmaterial1=pidmaterial or idmaterial2=pidmaterial);

select idmaterial2 into material2 from promocion where sysdate between promocion.fechainicio and promocion.fechafinal and (idmaterial1=pidmaterial or idmaterial2=pidmaterial);

select cantidad1 into cant1 from promocion where sysdate between promocion.fechainicio and promocion.fechafinal and (idmaterial1=pidmaterial or idmaterial2=pidmaterial);

select cantidad2 into cant2 from promocion where sysdate between promocion.fechainicio and promocion.fechafinal and (idmaterial1=pidmaterial or idmaterial2=pidmaterial);

select bono into bonus from promocion where sysdate between promocion.fechainicio and promocion.fechafinal and (idmaterial1=pidmaterial or idmaterial2=pidmaterial);

select sum(cantidad) into micant1 from estudiantematerial,promocion where idmaterial=material1 and idusuario=pidusuario and (fecharegistro between promocion.fechainicio and promocion.fechafinal);

select sum(cantidad) into micant2 from estudiantematerial,promocion where idmaterial=material2 and idusuario=pidusuario and (fecharegistro between promocion.fechainicio and promocion.fechafinal);

select tipo into tp from promocion where idpromocion=promo;

DBMS\_OUTPUT.PUT\_LINE(c);

DBMS\_OUTPUT.PUT\_LINE(c2);

DBMS\_OUTPUT.PUT\_LINE(tp);

DBMS\_OUTPUT.PUT\_LINE(micant1);

DBMS\_OUTPUT.PUT\_LINE(cant1);

DBMS\_OUTPUT.PUT\_LINE(micant2);

DBMS\_OUTPUT.PUT\_LINE(cant2);

DBMS\_OUTPUT.PUT\_LINE(bonus);

DBMS\_OUTPUT.PUT\_LINE(promo);

IF c>0 AND (c2)=0 THEN

IF tp='simple' AND micant1>=cant1 then

updatebono(pidusuario,bonus);

insertbono(pidusuario,promo);

ELSIF tp='doble' AND micant1>=cant1 AND micant2>=cant2 then

updatebono(pidusuario,bonus);

insertbono(pidusuario,promo);

end if;

end if;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END registrarmaterial;

---------------------------------------------------------------------------------------------------------------------------

create or replace PROCEDURE inserttelefono (ptelefono IN VARCHAR,

pidusuario IN INT)

AS

BEGIN

INSERT INTO telefono

(

telefono,

idusuario

)

VALUES

(

ptelefono,

pidusuario

);

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END inserttelefono;

---------------------------------------------------------------------------------------------------------------------------

create or replace PROCEDURE insertcorreo (pcorreo IN VARCHAR,

pidusuario IN INT)

AS

BEGIN

INSERT INTO correo

(

correoelectronico,

idusuario

)

VALUES

(

pcorreo,

pidusuario

);

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END insertcorreo;

---------------------------------------------------------------------------------------------------------------------------

CREATE OR replace PROCEDURE returnperfil (pidusuario IN INT,

c\_perfil OUT sys\_refcursor)

IS

BEGIN

OPEN c\_perfil FOR

select usuario.idusuario, usuario.nombre, usuario.apellido1, usuario.apellido2, estudiante.correo, estudiante.telefono, telefono.telefono as telefonosecundario, correo.correoelectronico as

correosecundario

from usuario

inner join estudiante on usuario.idusuario=pidusuario and usuario.idusuario=estudiante.idusuario

inner join telefono on telefono.idusuario=estudiante.idusuario

inner join correo on correo.idusuario=estudiante.idusuario;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END returnperfil;

---------------------------------------------------------------------------------------------------------------------------

CREATE OR replace PROCEDURE returnpromociones(piduniversidad IN INT,

c\_promo OUT sys\_refcursor)

IS

BEGIN

OPEN c\_promo FOR

SELECT idpromocion,iduniversidad,idmaterial1,idmaterial2,cantidad1,cantidad2,bono,fechainicio,fechafinal,tipo

FROM promocion

WHERE iduniversidad=piduniversidad;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END returnpromociones;

---------------------------------------------------------------------------------------------------------------------------

CREATE OR replace PROCEDURE deletepromocion(pidpromocion IN INT)

IS

BEGIN

DELETE

FROM promocion

WHERE idpromocion=pidpromocion;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END deletepromocion;

---------------------------------------------------------------------------------------------------------------------------

create or replace PROCEDURE inserttipocambio (ptipocambio IN NUMBER)

AS

BEGIN

UPDATE tipocambio

SET tipocambio.tipocambio = ptipocambio

WHERE tipocambio.idtipocambio=1;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END inserttipocambio;

--------------------------------------------------------------------------------------------------

CREATE OR replace PROCEDURE updatecolones (pidusuario IN INT,

pcantidad IN NUMBER)

AS

valor NUMBER;

cant NUMBER;

D\_n\_tecolones EXCEPTION;

BEGIN

SELECT tipocambio INTO valor FROM tipocambio WHERE idtipocambio=1;

SELECT cantidadtecolones INTO cant FROM estudiante WHERE idusuario=pidusuario;

IF pcantidad>cant THEN

RAISE D\_n\_tecolones;

ELSE

UPDATE estudiante

SET estudiante.cantidadcolones = estudiante.cantidadcolones+pcantidad\*valor,

estudiante.cantidadtecolones = estudiante.cantidadtecolones-pcantidad

WHERE estudiante.idusuario=pidusuario;

END IF;

COMMIT;

END updatecolones;

---------------------------------------------------------------------------------------------------

CREATE OR replace PROCEDURE returnestudiante(pidusuario IN INT,

c\_sede OUT sys\_refcursor)

IS

BEGIN

OPEN c\_sede FOR

SELECT usuario.idusuario,usuario.nombre, usuario.apellido1, usuario.apellido2, estudiante.descripcion, estudiante.correo, estudiante.telefono, estudiante.cantidadcolones,

estudiante.cantidadtecolones, estado.estado, universidad.nombre as universidad, sede.nombre as sede

FROM usuario,estudiante,sede,estado, universidad

WHERE usuario.idusuario=pidusuario AND usuario.idusuario=estudiante.idusuario AND estudiante.idestado=estado.idestado AND estudiante.idsede=sede.idsede AND sede.iduniversidad=universidad.iduniversidad;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END returnestudiante;

---------------------------------------------------------------------------------------------------

create or replace PROCEDURE returnacopio(piduniversidad IN INT,

c\_acopio OUT sys\_refcursor)

IS

BEGIN

OPEN c\_acopio FOR

SELECT centroacopio.idcentroacopio, centroacopio.nombre

FROM centroacopio,sede

WHERE sede.iduniversidad=piduniversidad and sede.idsede=centroacopio.idsede;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

END returnacopio;