PROCEDIMIENTOS ALMACENADOS

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

create or replace procedure DELETE\_CLIENT(deleterut in varchar2, resultado OUT INTEGER)

is

total number;

begin

total :=0;

select count(\*) into total from users where RUT = deleterut;

if total = 0 then

resultado := 0;

else

delete FROM users where RUT = deleterut;

resultado :=1;

end if;

commit;

END;

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

create or replace procedure DELETE\_DEPARTMENT(deleteId in INTEGER, resultado OUT INTEGER)

is

totalDepartment number;

totalInventory number;

begin

totalDepartment :=0;

totalinventory :=0;

select count(\*) into totalDepartment from department where ID = deleteId;

select count(\*) into totalInventory from inventory where department\_id = deleteId;

/\*SI EL DEPARTAMENTO SE ENCUENTRA Y NO POSEE INVENTARIO\*/

if totalDepartment = 1 AND totalinventory = 0 then

delete FROM department where ID = deleteId;

resultado := 1;

end if;

if totalinventory > 0 then

resultado := 2;

end if;

commit;

END;

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

create or replace procedure DELETE\_DISCOUNT(deleteID in INTEGER, resultado OUT INTEGER)

is

totalDiscount number;

totalRegistration number;

begin

totalRegistration :=0;

totalDiscount :=0;

select count(\*) into totalDiscount from discount where id = deleteID;

select count(\*) into totalRegistration from registration where discount\_id = deleteID;

if totalregistration = 0 and totalDiscount = 1 then

delete from discount where id = deleteid;

resultado :=1;

end if;

if totalregistration > 0 then

resultado :=2;

end if;

commit;

END;

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

create or replace procedure DELETE\_DRIVER(delRut in varchar2, resultado OUT INTEGER)

is

totalDriver number;

totalTransport number;

begin

totalDriver :=0;

totalTransport := 0;

select count(\*) into totalDriver from driver where RUT = delRut;

select count(\*) into totalTransport from transport where driver\_rut = delRut;

if totaldriver = 1 and totalTransport = 0 then

delete FROM driver where RUT = delRut;

resultado :=1;

end if;

if totaltransport > 0 then

resultado :=2;

end if;

commit;

END;

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

create or replace PROCEDURE DELETE\_INVENTORY(inID IN INTEGER, resultado OUT INTEGER)

IS

total number;

BEGIN

total := 0;

select count(\*) into total from inventory where id = inID;

if total = 1 then

delete from inventory where id = inID;

resultado :=1;

else

resultado :=0;

end if;

commit;

END;

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

create or replace procedure DELETE\_TOUR(delID in INTEGER, resultado OUT INTEGER)

is

totalTour number;

totalReservation number;

begin

totalTour :=0;

totalreservation := 0;

select count(\*) into totalTour from tour where id = delID;

select count(\*) into totalreservation from registration where tour\_id = delID;

/\*SI EL TOUR EXISTE Y NO HAY UN REGISTRO EN LA TABLA REGISTRATION SE PUEDE BORRAR\*/

if totalTour > 0 and totalreservation = 0 then

delete FROM tour where id = delID;

resultado := 1;

end if;

if totalreservation > 0 then

resultado :=2;

end if;

commit;

END;

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

create or replace procedure DELETE\_TRANSPORT(deleteId in INTEGER, resultado OUT INTEGER)

is

totalTransport number;

totalRgistration number;

begin

totalTransport :=0;

totalRgistration :=0;

select count(\*) into totalTransport from transport where ID = deleteId;

select count(\*) into totalRgistration from registration where transport\_id = deleteId;

/\*SI EL DEPARTAMENTO SE ENCUENTRA Y NO POSEE INVENTARIO\*/

if totalTransport = 1 AND totalRgistration = 0 then

delete FROM transport where ID = deleteId;

resultado := 1;

end if;

if totalRgistration > 0 then

resultado := 2;

end if;

commit;

END;

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

create or replace PROCEDURE DELETE\_VEHICLE(inPatent IN VARCHAR2, resultado OUT INTEGER)

IS

totalVehicle number;

totalTransport number;

BEGIN

totalVehicle := 0;

totalTransport := 0;

select count(\*) into totalVehicle from vehicle where patent = inPatent;

select count(\*) into totalTransport from transport where vehicle\_patent = inPatent;

/\*SI NO SE ENCUENTRAN VEHICULOS CON ESA PATENTE\*/

if totalTransport = 0 and totalVehicle = 1 then

delete from vehicle where patent = inPatent;

resultado := 1;

end if;

if totalTransport > 0 then

resultado :=2;

end if;

commit;

END;

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

create or replace procedure INSERT\_CLIENT(nrut in varchar2,name in varchar2,lastnamef in varchar2,lastnamem in varchar2, usertype in varchar2,password in varchar2, resultado OUT INTEGER)

is

total number;

begin

total :=0;

select count(\*) into total from users where RUT = nrut;

if total = 0 then

insert into USERS values (nrut, name, lastnamef, lastnamem, usertype, password);

resultado := 1;

else

resultado :=0;

end if;

commit;

end;

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

create or replace procedure INSERT\_DEPARTMENT(region in varchar2, address in varchar2, floor in varchar2, price in INT, state in varchar2, dirr in INT)

is

begin

insert into department values (sequence\_department.nextval, region, address, floor, price, state, dirr);

commit;

end;

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

create or replace procedure INSERT\_DISCOUNT(addName IN VARCHAR2, addPercent IN INTEGER)

is

begin

insert into discount values (sequence\_discount.nextval, addName, addPercent);

commit;

end;

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

create or replace procedure INSERT\_DRIVER(newRut in varchar2,newnName in varchar2,newLastNameF in varchar2,newlastNameM in varchar2, resultado OUT INTEGER)

is

total number;

begin

total :=0;

select count(\*) into total from driver where RUT = newRut;

if total = 0 then

insert into driver values (newRut, newnName, newLastNameF, newlastNameM);

resultado := 1;

dbms\_output.put\_line('Conductor ingresado correctamente');

else

resultado :=0;

dbms\_output.put\_line('El conductor no se pudo ingresar');

end if;

commit;

end;

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

create or replace procedure INSERT\_INVENTORY(name in varchar2, quantity in INTEGER, description IN VARCHAR2, price IN INTEGER, department\_id IN INTEGER, image IN BLOB)

is

begin

insert into inventory values (sequence\_inventory.nextval, name, quantity, description, price, department\_id, image);

commit;

end;

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

create or replace procedure INSERT\_TOUR(addOrigin in varchar2,addDestination in varchar2, addDepartureHour in varchar2, addArrivalHour in varchar2, addPrice in INTEGER, addDateArrival in VARCHAR2, addDepartureDate in Varchar2)

is

begin

insert into tour values (sequence\_tour.nextval, addOrigin, addDestination, addDepartureHour, addArrivalHour, addPrice, to\_date(addDateArrival, 'dd/MM/yyyy'), to\_date(addDepartureDate, 'dd/MM/yyyy'));

commit;

end;

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

create or replace procedure INSERT\_TRANSPORT(addOrigen in varchar2, addDestination in varchar2, addDepartureHour in varchar2, addArrivalHour in varchar2, addPrice in INTEGER, addDriverRut in VARCHAR2, addVehiclePatent in varchar2, addDateTransport in varchar2)

is

begin

insert into transport values (sequence\_transport.nextval, addOrigen, addDestination, addDepartureHour, addArrivalHour, addPrice, addDriverRut, addVehiclePatent, to\_date(addDateTransport, 'dd/MM/yyyy'));

commit;

end;

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

create or replace procedure INSERT\_VEHICLE(inPatent in varchar2,inColor in varchar2,inBrand in varchar2,inModel in varchar2, inYear in INTEGER, resultado OUT INTEGER)

is

total number;

begin

total :=0;

select count(\*) into total from vehicle where patent = inPatent;

if total = 0 then

insert into vehicle values (inPatent, inColor, inBrand, inModel, inYear);

resultado :=1;

else

resultado :=0;

end if;

commit;

end;

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

create or replace PROCEDURE LISTAR\_CLIENTS(OUT\_CURSOR OUT SYS\_REFCURSOR)

AS

BEGIN

OPEN out\_cursor FOR SELECT rut, username, lastnamef, lastnamem,usertype,pass FROM USERS WHERE usertype = 'cliente';

END;

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

create or replace PROCEDURE LISTAR\_DEPARTMENTS(OUT\_CURSOR OUT SYS\_REFCURSOR)

AS

BEGIN

OPEN out\_cursor FOR SELECT id, region, address, floor, price, state, dirr FROM department;

END;

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

create or replace PROCEDURE LISTAR\_DISCOUNTS(OUT\_CURSOR OUT SYS\_REFCURSOR)

AS

BEGIN

OPEN out\_cursor FOR SELECT id, name, percent FROM discount;

END;

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

create or replace PROCEDURE LISTAR\_DRIVERS(OUT\_CURSOR OUT SYS\_REFCURSOR)

AS

BEGIN

OPEN out\_cursor FOR SELECT rut, name, lastnamef, lastnamem FROM DRIVER;

END;

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

create or replace PROCEDURE LISTAR\_IDDEPARTMENT(OUT\_CURSOR OUT SYS\_REFCURSOR)

AS

BEGIN

OPEN out\_cursor FOR SELECT ID FROM department;

END;

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

create or replace PROCEDURE LISTAR\_INVENTORY(OUT\_CURSOR OUT SYS\_REFCURSOR)

AS

BEGIN

OPEN out\_cursor FOR SELECT id, name,quantity,description, price, department\_id, image FROM inventory;

END;

------------------------------------------------------------------------------------------------------------------------create or replace PROCEDURE LISTAR\_PATENTVEHICLE(OUT\_CURSOR OUT SYS\_REFCURSOR)

AS

BEGIN

OPEN out\_cursor FOR SELECT patent FROM vehicle;

END;

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

create or replace PROCEDURE LISTAR\_RUTDRIVER(OUT\_CURSOR OUT SYS\_REFCURSOR)

AS

BEGIN

OPEN out\_cursor FOR SELECT RUT FROM DRIVER;

END;

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

create or replace PROCEDURE LISTAR\_TOURS(OUT\_CURSOR OUT SYS\_REFCURSOR)

AS

BEGIN

OPEN out\_cursor FOR SELECT ID, origin, destination, departurehour, arrivaltime, priceperperson, datearrival, datedeparture FROM TOUR;

END;

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

create or replace PROCEDURE LISTAR\_TRANSPORT(OUT\_CURSOR OUT SYS\_REFCURSOR)

AS

BEGIN

OPEN out\_cursor FOR SELECT id, origin, destination, departureHour, arrivalHour, price, driver\_rut, vehicle\_patent, datetransport FROM TRANSPORT;

END;

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

create or replace PROCEDURE LISTAR\_VEHICLES(OUT\_CURSOR OUT SYS\_REFCURSOR)

AS

BEGIN

OPEN out\_cursor FOR SELECT PATENT, COLOR, BRAND, MODEL, YEAR FROM VEHICLE;

END;

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

create or replace procedure LOGIN\_USERS(rutIngresado in varchar2, passwords in varchar2, tipoUsuario out varchar2, clave out varchar2, cantidad OUT NUMBER)

is

total number;

begin

total :=0;

SELECT COUNT(\*) INTO total FROM USERS where rut = rutIngresado and pass = passwords;

if (total =0) then

cantidad := 0;

dbms\_output.put\_line('El usuario no existe');

else

SELECT usertype,pass into tipoUsuario,clave from USERS where rut = rutIngresado and pass = passwords;

cantidad := 1;

end if;

end;

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

create or replace procedure READ\_CLIENT(readRut IN VARCHAR2,OUT\_CURSOR OUT SYS\_REFCURSOR)

is

total number;

begin

OPEN out\_cursor FOR SELECT rut, username, lastNameF, lastNameM FROM users where usertype = 'cliente' and rut like '%'||readRut||'%';

END;

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

create or replace procedure READ\_DEPARTMENTS(readRegion in varchar2, readStatus IN varchar2,OUT\_CURSOR OUT SYS\_REFCURSOR)

is

total number;

begin

OPEN out\_cursor FOR SELECT id, region, address, floor, price, state, dirr FROM department where region = readRegion and state = readStatus;

END;

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

create or replace procedure READ\_DISCOUNT(readName in varchar2, OUT\_CURSOR OUT SYS\_REFCURSOR)

is

begin

OPEN out\_cursor FOR SELECT id, name, percent FROM discount where name like '%'||readName||'%';

END;

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

create or replace procedure READ\_DRIVER(readRut IN VARCHAR2,OUT\_CURSOR OUT SYS\_REFCURSOR)

is

total number;

begin

OPEN out\_cursor FOR SELECT rut, name, lastNameF, lastNameM FROM DRIVER where rut like '%'||readRut||'%';

END;

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

create or replace procedure READ\_INVENTORY(readID in INTEGER,OUT\_CURSOR OUT SYS\_REFCURSOR)

is

total number;

begin

OPEN out\_cursor FOR SELECT id, name, quantity, description, price, department\_id, image

FROM inventory where department\_id = readID;

END;

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

create or replace procedure READ\_VEHICLE(readPatente IN VARCHAR2,OUT\_CURSOR OUT SYS\_REFCURSOR)

is

total number;

begin

OPEN out\_cursor FOR SELECT PATENT, COLOR, BRAND, MODEL, YEAR FROM VEHICLE where patent like '%'||readPatente||'%';

END;

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

create or replace procedure UPDATE\_CLIENT(newRut in varchar2,newnName in varchar2,newLastNameF in varchar2,newlastNameM in varchar2, resultado OUT INTEGER)

is

total number;

begin

total :=0;

select count(\*) into total from users where RUT = newRut;

if total = 1 then

UPDATE users SET username = newnName,

LASTNAMEF = newLastNameF,

lASTNAMEM = newlastNameM

WHERE RUT = newRut;

resultado :=1;

else

resultado :=0;

end if;

commit;

end ;

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

create or replace procedure UPDATE\_DEPARTMENT(newId IN INTEGER,newRegion in varchar2,newAddress in varchar2,newFloor in varchar2,newPrice in INTEGER, newState IN VARCHAR2,newDirr IN INTEGER, resultado OUT INTEGER)

is

total number;

begin

total :=0;

select count(\*) into total from department where ID = newId;

if total = 1 then

UPDATE department SET region = newRegion,

address = newAddress,

floor = newFloor,

price = newPrice,

state = newState,

dirr = newDirr

WHERE ID = newId;

resultado :=1;

else

resultado :=0;

end if;

commit;

end ;

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

create or replace procedure UPDATE\_DISCOUNT(newID in INTEGER, newName in varchar2, newPercent in INTEGER,resultado OUT INTEGER)

is

total number;

begin

total :=0;

select count(\*) into total from discount where ID = newID;

if total = 1 then

UPDATE Discount SET name = newname,

percent = newpercent

WHERE ID = newId;

resultado :=1;

else

resultado :=0;

end if;

commit;

end ;

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

create or replace procedure UPDATE\_DRIVER(newRut in varchar2,newnName in varchar2,newLastNameF in varchar2,newlastNameM in varchar2, resultado OUT INTEGER)

is

total number;

begin

total :=0;

select count(\*) into total from driver where RUT = newRut;

if total = 1 then

UPDATE DRIVER SET NAME = newnName,

LASTNAMEF = newLastNameF,

lASTNAMEM = newlastNameM

WHERE RUT = newRut;

resultado :=1;

else

resultado :=0;

end if;

commit;

end ;

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

create or replace procedure UPDATE\_INVENTORY(newId IN INTEGER, newName in varchar2, newQuantity in INTEGER, newDescription in varchar2, newPrice in integer,newDepartment\_id IN INTEGER, resultado OUT INTEGER)

is

total number;

begin

total :=0;

select count(\*) into total from inventory where ID = newId;

if total = 1 then

UPDATE inventory SET name = newName,

quantity = newQuantity,

description = newDescription,

price = newPrice,

department\_id = newDepartment\_id

WHERE id = newId;

resultado :=1;

else

resultado :=0;

end if;

commit;

end ;

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

create or replace procedure UPDATE\_TOUR(newID in INTEGER,newOrigin in varchar2, newDestination in varchar, newDepartureHour in varchar2, newArrivalTime in varchar2,newPricePerPerson in INTEGER, newDateArrival in varchar2 , newDateDeparture in varchar2,resultado OUT INTEGER)

is

total number;

begin

total :=0;

select count(\*) into total from tour where ID = newID;

if total = 1 then

UPDATE TOUR SET origin = neworigin,

destination = newdestination,

departurehour = newdeparturehour,

arrivaltime = newarrivaltime,

priceperperson = newpriceperperson,

datearrival = to\_date(newdatearrival, 'dd/MM/yyyy'),

datedeparture = to\_date(newdatedeparture, 'dd/MM/yyyy')

WHERE ID = newId;

resultado :=1;

else

resultado :=0;

end if;

commit;

end ;

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

create or replace procedure UPDATE\_TRANSPORT(newID in INTEGER,newOrigin in varchar2, newDestination in varchar, newDepartureHour in varchar2, newArrivalHour in varchar2,newPrice in INTEGER, newDriverRut in varchar2 , newPatentVehicle in varchar2,newDateTransport in varchar2,resultado OUT INTEGER)

is

total number;

begin

total :=0;

select count(\*) into total from transport where ID = newID;

if total = 1 then

UPDATE transport SET origin = neworigin,

destination = newdestination,

departurehour = newdeparturehour,

arrivalhour = newArrivalHour,

price = newPrice,

driver\_rut = newDriverRut,

vehicle\_patent = newPatentVehicle,

datetransport = to\_date(newDateTransport, 'dd/MM/yyyy')

WHERE ID = newId;

resultado :=1;

else

resultado :=0;

end if;

commit;

end ;

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

create or replace procedure UPDATE\_VEHICLE(newPatent in varchar2, newBrand in varchar2, newColor in varchar2, newModel in varchar2, newYear in INTEGER, resultado OUT INTEGER)

is

total number;

begin

total:=0;

select count(\*)into total from vehicle where patent = newPatent;

if total= 1 then

UPDATE vehicle SET

brand = newBrand,

color = newColor,

model = newModel,

year = newYear

where patent = newPatent;

resultado := 1;

else

resultado :=0;

end if;

END;

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

create or replace procedure READ\_TRANSPORT(readDate IN VARCHAR2,OUT\_CURSOR OUT SYS\_REFCURSOR)

is

total number;

begin

OPEN out\_cursor FOR SELECT id, origin,destination, departurehour, arrivalhour, price, driver\_rut, vehicle\_patent, datetransport FROM transport where datetransport = TO\_DATE(readDate, 'dd/MM/yyyy');

END;

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

create or replace procedure READ\_TOUR(readDate IN VARCHAR2,OUT\_CURSOR OUT SYS\_REFCURSOR)

is

total number;

begin

OPEN out\_cursor FOR SELECT id, origin,destination, departurehour, arrivaltime, priceperperson, datearrival, datedeparture FROM TOUR where datedeparture = TO\_DATE(readDate, 'dd/MM/yyyy');

END;

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

create or replace procedure READ\_USERS(readRut IN VARCHAR2,OUT\_CURSOR OUT SYS\_REFCURSOR)

is

total number;

begin

OPEN out\_cursor FOR SELECT rut, username, lastNameF, lastNameM, usertype FROM users where usertype <> 'cliente' and rut like '%'||readRut||'%';

END;

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

create or replace PROCEDURE LISTAR\_USERS(OUT\_CURSOR OUT SYS\_REFCURSOR)

AS

BEGIN

OPEN out\_cursor FOR SELECT rut, username, lastnamef, lastnamem,usertype, pass FROM USERS WHERE usertype <> 'cliente';

END;