**Universidad Tecnológica de Panamá**

**Facultad de Ingeniería de Sistemas Computacionales**

**Licenciatura en Ingeniería de Sistemas de Información**

**Departamento de Sistemas de Información**

Sistemas de Bases de Datos II

Equipo #2

Laboratorio #7

Facilitador:

Ing. Henry Lezcano

Integrantes grupo 2:

Karen Cabrera 20-14-5403

Medardo Logreira 8-879-1679

Sebastian Zamora 20-15-4392

Georlenys Martinez 8-926-2256

Grupo:

1IF131

II Semestre, 2020

**UNIVERSIDAD TECNOLOGICA DE PANAMA**

**FACULTAD DE INGENIERIA DE SISTEMAS COMPUTACIONES**

**LICENCIATURA EN INGENIERIA DE SISTEMAS DE INFORMACION**

**SISTEMAS DE BASE DE DATOS II**

**LABORATORIO NO.7**

**FUNDAMENTOS DEL LENGUAJE-CURSORES-PROCEDIMIENTOS Y FUNCIONES**

**y TRIGGERS ‘APLICAR PROGRAMACION ALMACENADA DE BASE DE DATOS**

**PL/SQL ORACLE’**

Create Table Sucursal (

upd\_sucursal Number,

Cod\_Sucursal Varchar2 (2) Primary Key not null,

Cod\_Ahorro Varchar2 (2),

monto\_ahor\_v Number,

monto\_ahor\_n Number,

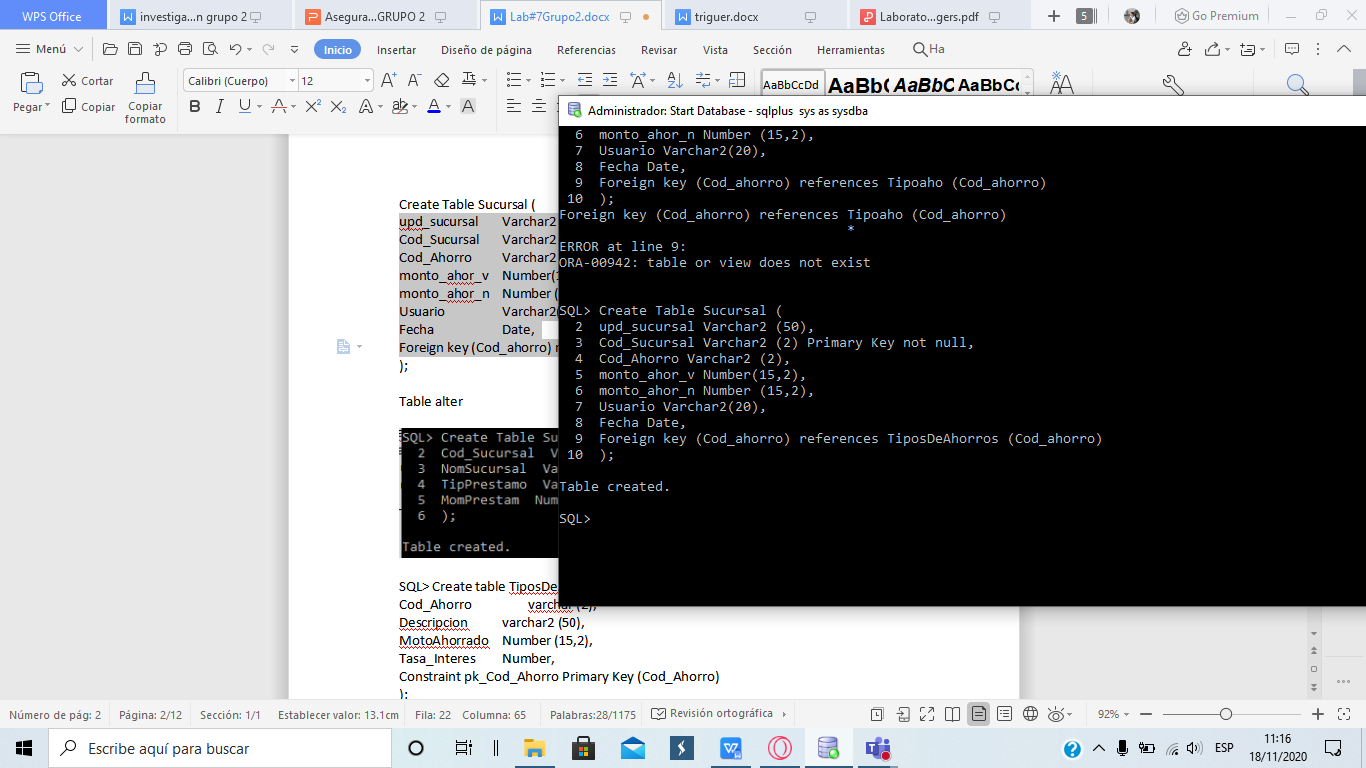
Usuario Varchar2(20),

Fecha Date,

Foreign key (Cod\_ahorro) references TiposDeAhorros (Cod\_ahorro)

);

Table alter



SQL> Create table TiposDeAhorros (

Cod\_Ahorro varchar (2),

Descripcion varchar2 (50),

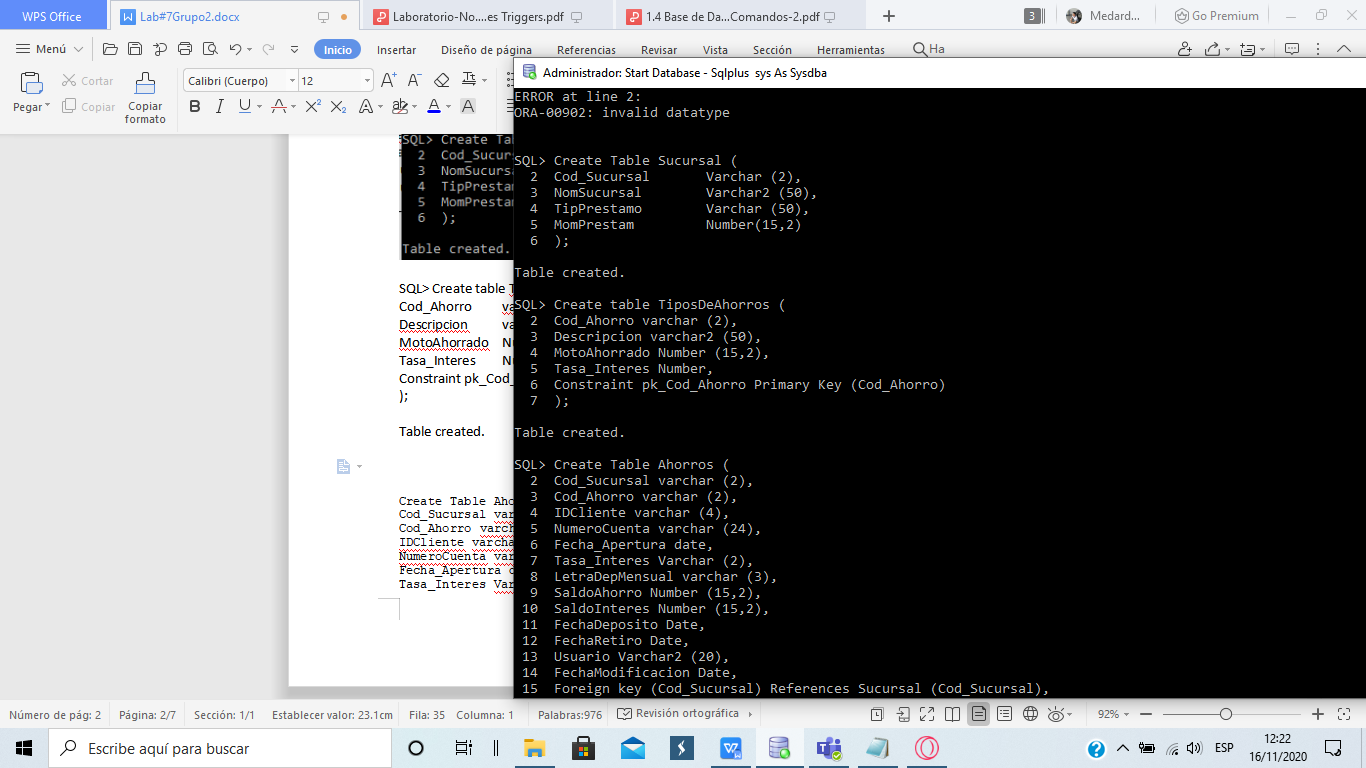
MotoAhorrado Number,

Tasa\_Interes Number,

Constraint pk\_Cod\_Ahorro Primary Key (Cod\_Ahorro)

);

Table created.



Create Table Ahorros (

Cod\_Sucursal varchar (2),

Cod\_Ahorro varchar (2),

IDCliente varchar (4),

NumeroCuenta varchar (24),

Fecha\_Apertura date,

Tasa\_Interes Varchar (2),

LetraDepMensual varchar (3),

SaldoAhorro Number,

SaldoInteres Number,

FechaDeposito Date,

FechaRetiro Date,

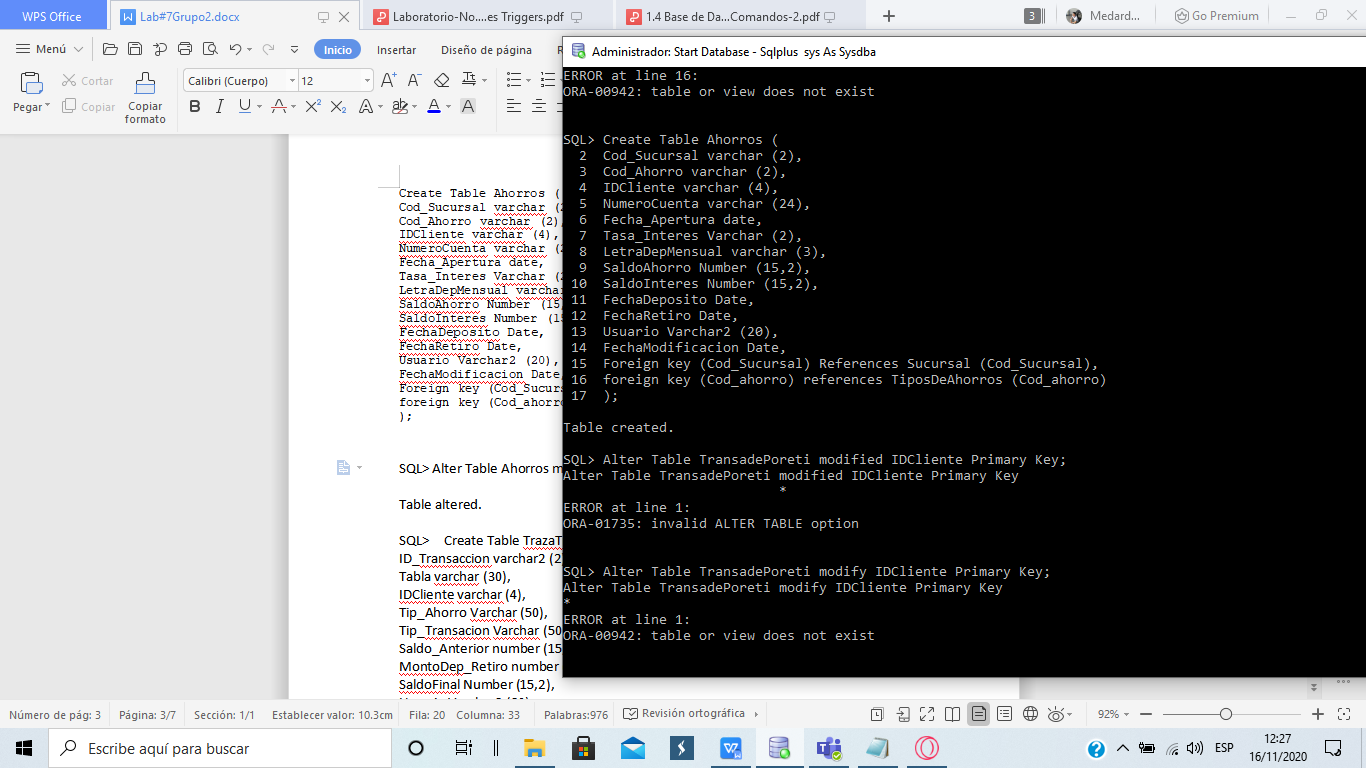
Usuario Varchar2 (20),

FechaModificacion Date,

Foreign key (Cod\_Sucursal) References Sucursal (Cod\_Sucursal),

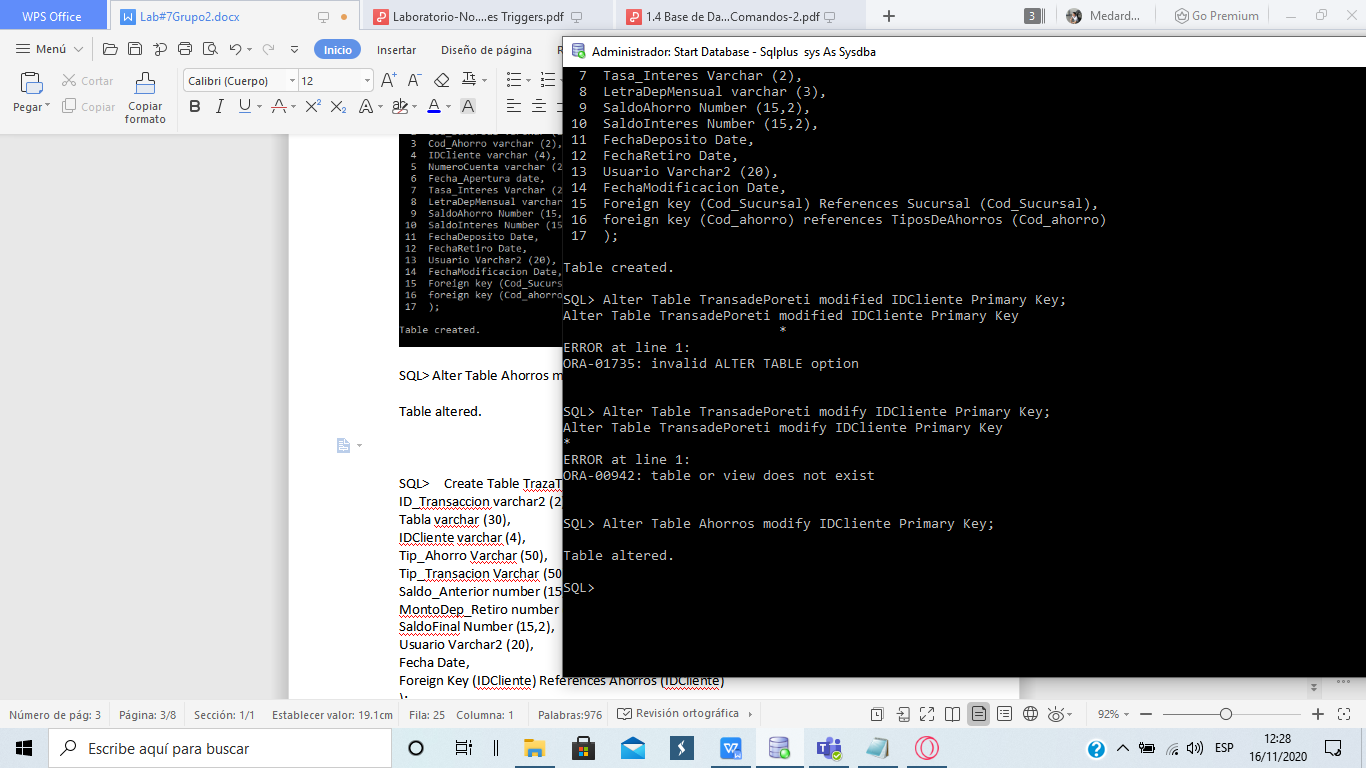
foreign key (Cod\_ahorro) references TiposDeAhorros (Cod\_ahorro)

);



SQL> Alter Table Ahorros modify IDCliente Primary Key;

Table altered.



SQL> Create Table TrazaTransa (

ID\_Transaccion varchar2 (2) Primary Key,

Tabla varchar (30),

IDCliente varchar (4),

Tip\_Ahorro Varchar (50),

Tip\_Transacion Varchar (50),

Saldo\_Anterior number,

MontoDep\_Retiro number,

SaldoFinal Number,

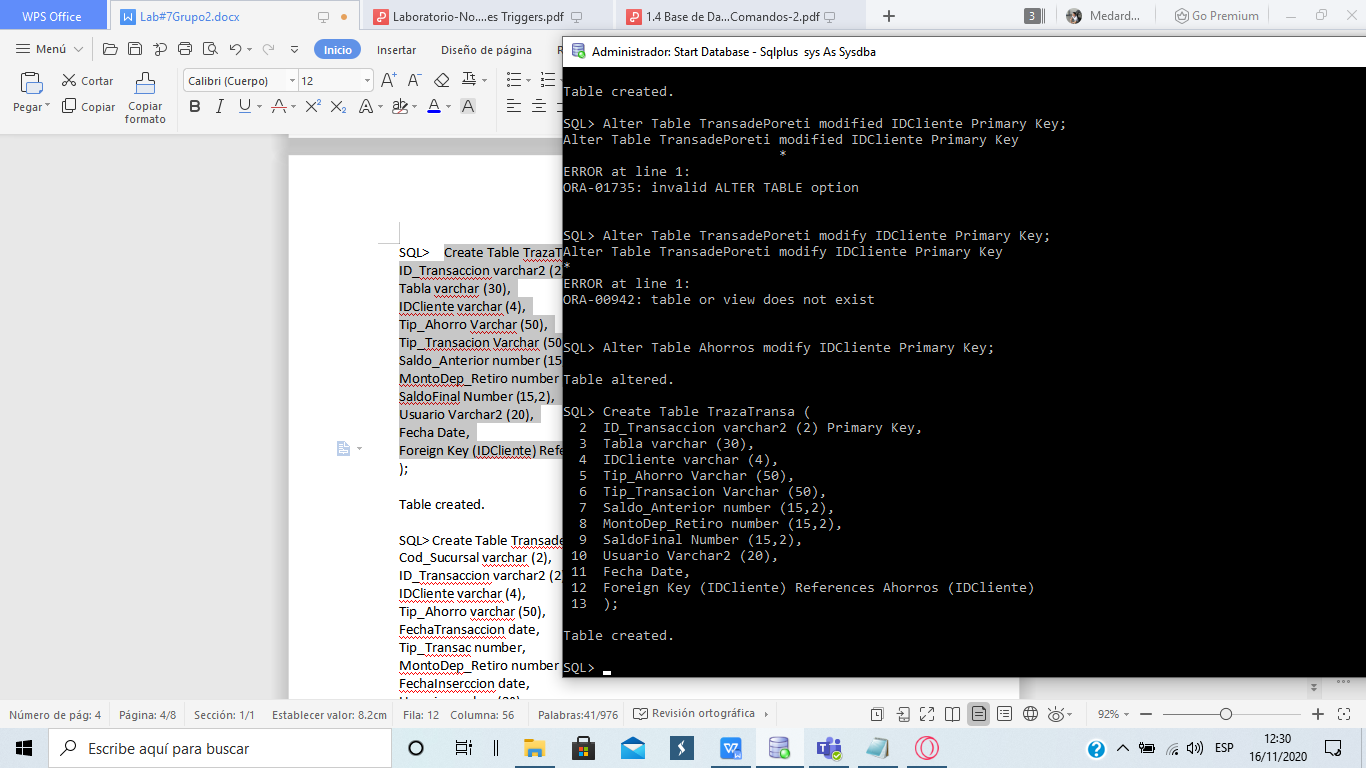
Usuario Varchar2 (20),

Fecha Date,

Foreign Key (IDCliente) References Ahorros (IDCliente)

);

Table created.



SQL> Create Table TransadePoreti (

Cod\_Sucursal varchar (2),

ID\_Transaccion varchar2 (2),

IDCliente varchar (4),

Tip\_Ahorro varchar (50),

FechaTransaccion date,

Tip\_Transac number,

MontoDep\_Retiro number,

FechaInserccion date,

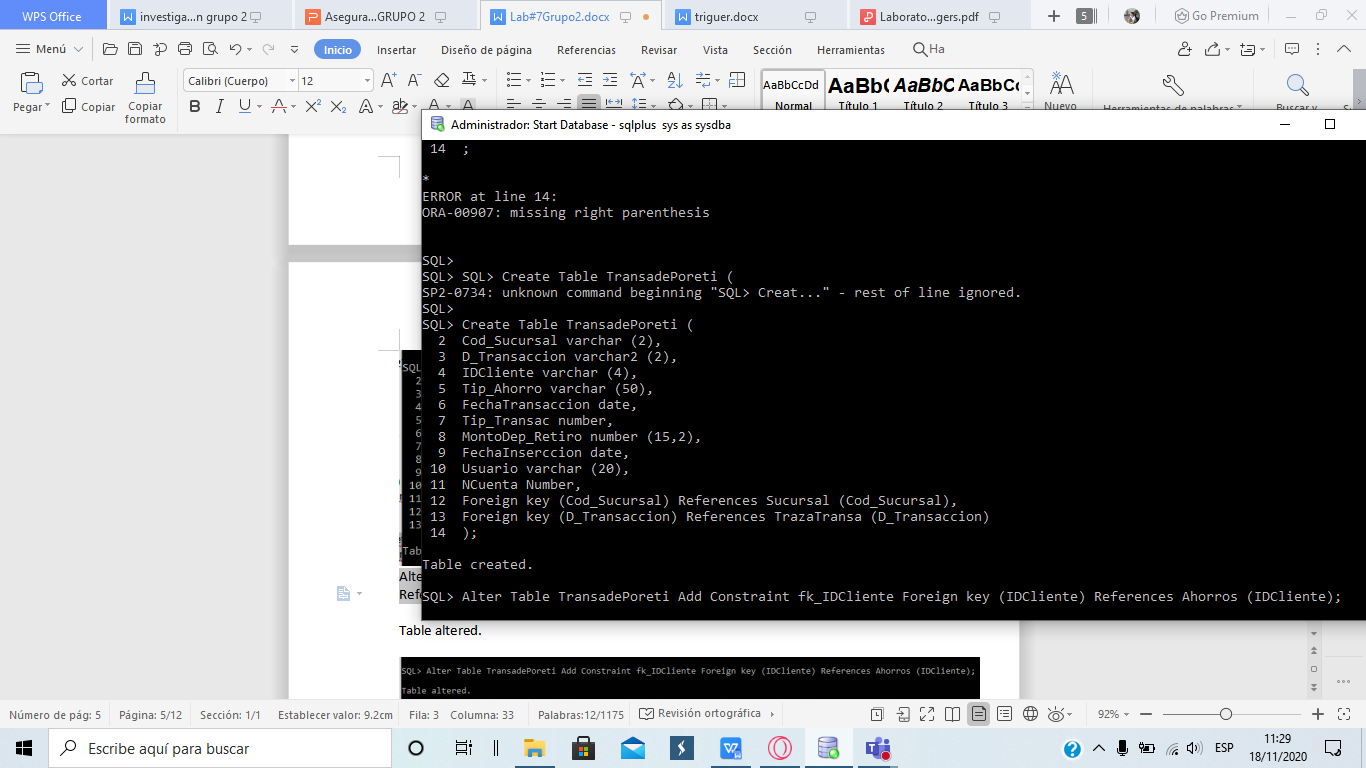
Usuario varchar (20),

NCuenta Number,

Foreign key (Cod\_Sucursal) References Sucursal (Cod\_Sucursal),

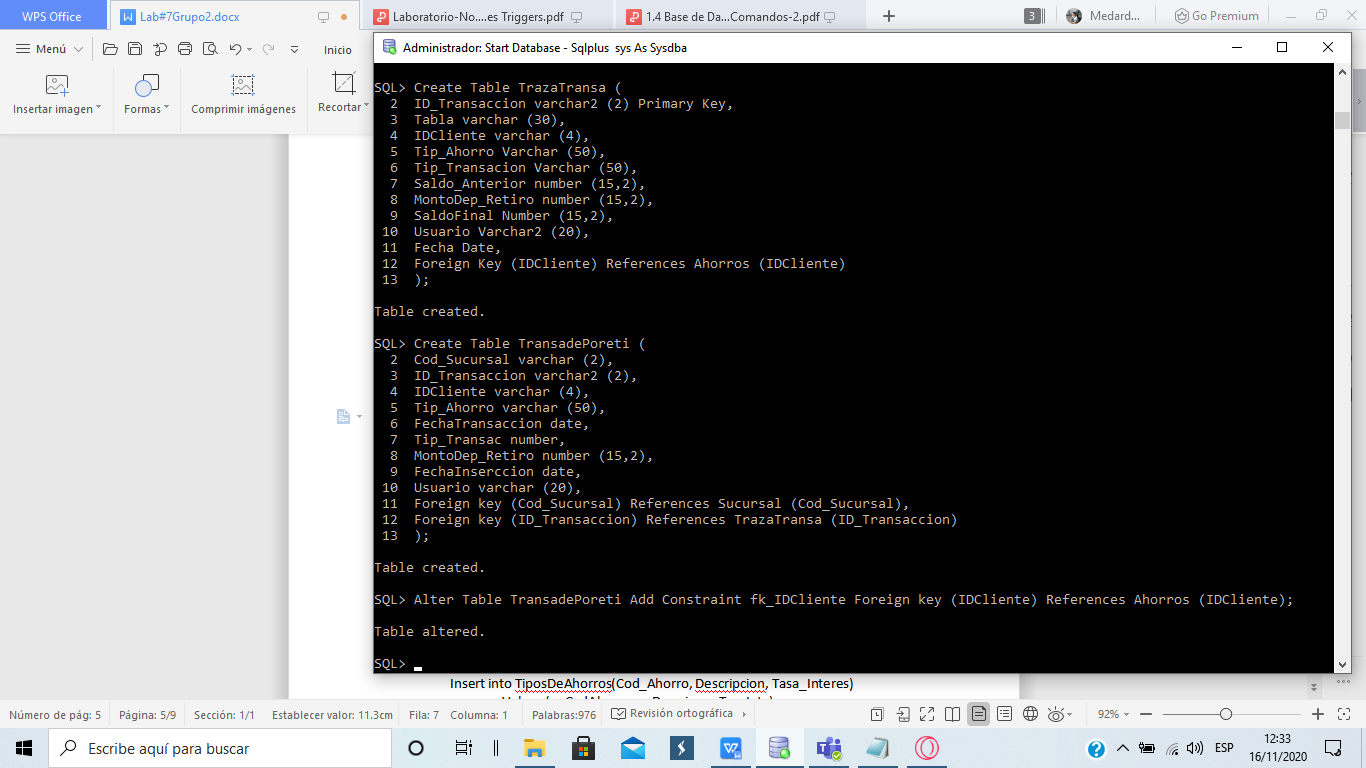
Foreign key (ID\_Transaccion) References TrazaTransa (ID\_Transaccion)

);



Alter Table TransadePoreti Add Constraint fk\_IDCliente Foreign key (IDCliente) References Ahorros (IDCliente);

Table altered.



Procedimientos

**Procedimiento** almacenado para la carga o inserción de la tabla paramétrica

* Tipo de ahorros

SQL> Create Or Replace Procedure InserTipAhorro (

p\_CodAho IN TiposDeAhorros.Cod\_Ahorro%Type,

p\_Descri IN TiposDeAhorros.Descripcion%Type,

p\_MonAho IN TiposDeAhorros.MotoAhorrado%Type,

p\_TasInt IN TiposDeAhorros.Tasa\_Interes%Type,

p\_Error OUT Varchar2) AS

Begin

Insert into TiposDeAhorros (Cod\_Ahorro, Descripcion, MotoAhorrado, Tasa\_Interes)

Values (p\_CodAho, p\_Descri, p\_MonAho, p\_TasInt);

p\_Error := 'Ahorro Registrado';

Exception

When Others Then

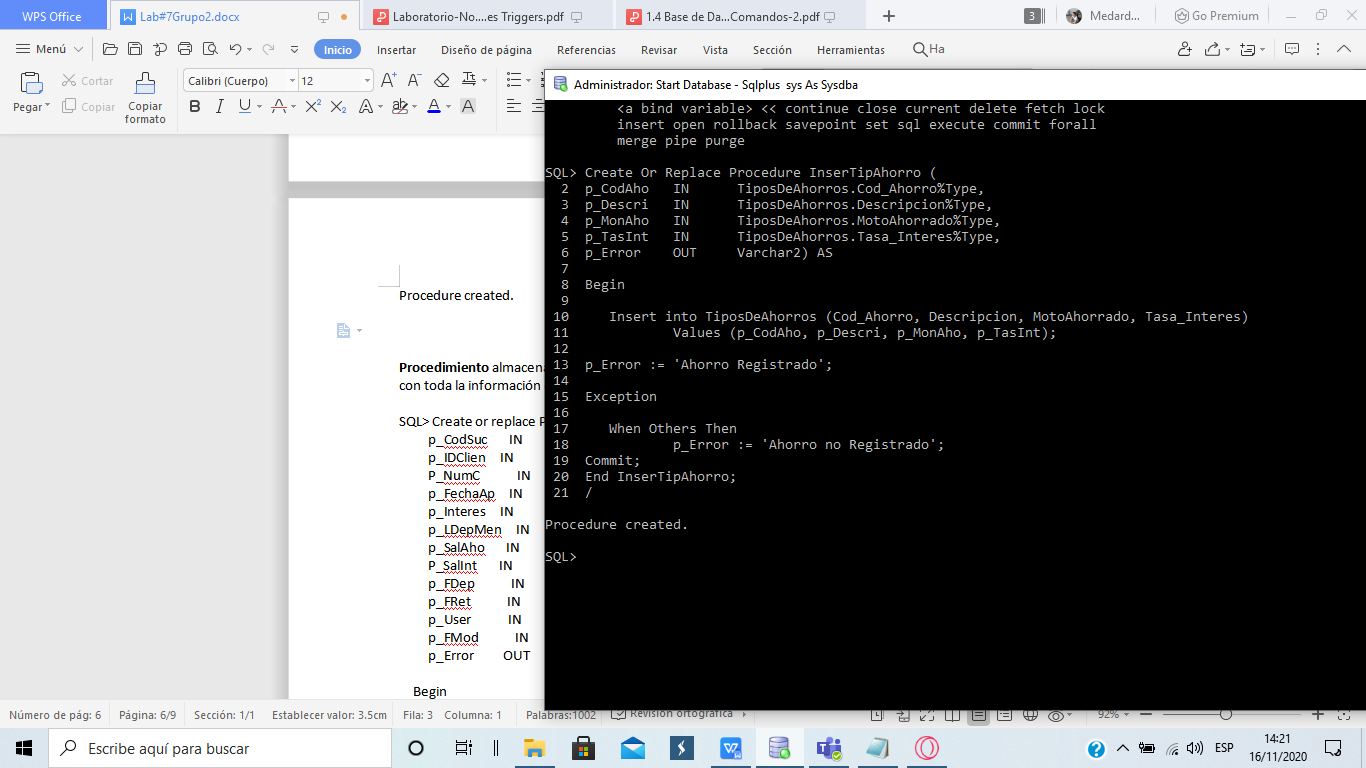
p\_Error := 'Ahorro no Registrado';

Commit;

End InserTipAhorro;

/

Procedure created.



**Procedimiento** almacenado para la apertura o inserción de los ahorros aprobados

con toda la información correspondiente.

SQL> Create or replace Procedure InserAhorros(

p\_CodSuc IN Ahorros.Cod\_Sucursal%Type,

p\_IDClien IN Ahorros.IDCliente%Type,

P\_NumC IN Ahorros.NumeroCuenta%Type,

p\_FechaAp IN Ahorros.Fecha\_Apertura%Type,

p\_Interes IN Ahorros.Tasa\_Interes%Type,

p\_LDepMen IN Ahorros.LetraDepMensual%Type,

p\_SalAho IN Ahorros.SaldoAhorro%Type,

P\_SalInt IN Ahorros.SaldoInteres%Type,

p\_FDep IN Ahorros.FechaDeposito%Type,

p\_FRet IN Ahorros.FechaRetiro%Type,

p\_User IN Ahorros.Usuario%Type,

p\_FMod IN Ahorros.FechaModificacion%Type,

p\_Error OUT Varchar2) AS

Begin

Insert into Ahorros (Cod\_Sucursal, IDCliente, NumeroCuenta, Fecha\_Apertura, Tasa\_Interes, LetraDepMensual, SaldoAhorro, SaldoInteres, FechaDeposito, FechaRetiro, Usuario, FechaModificacion)

Values (p\_CodSuc, p\_IDClien, P\_NumC, p\_FechaAp, p\_Interes, p\_LDepMen, p\_SalAho, P\_SalInt, p\_FDep, p\_FRet, p\_User, p\_FMod);

p\_Error := 'Apertura creada';

Exception

When Others Then

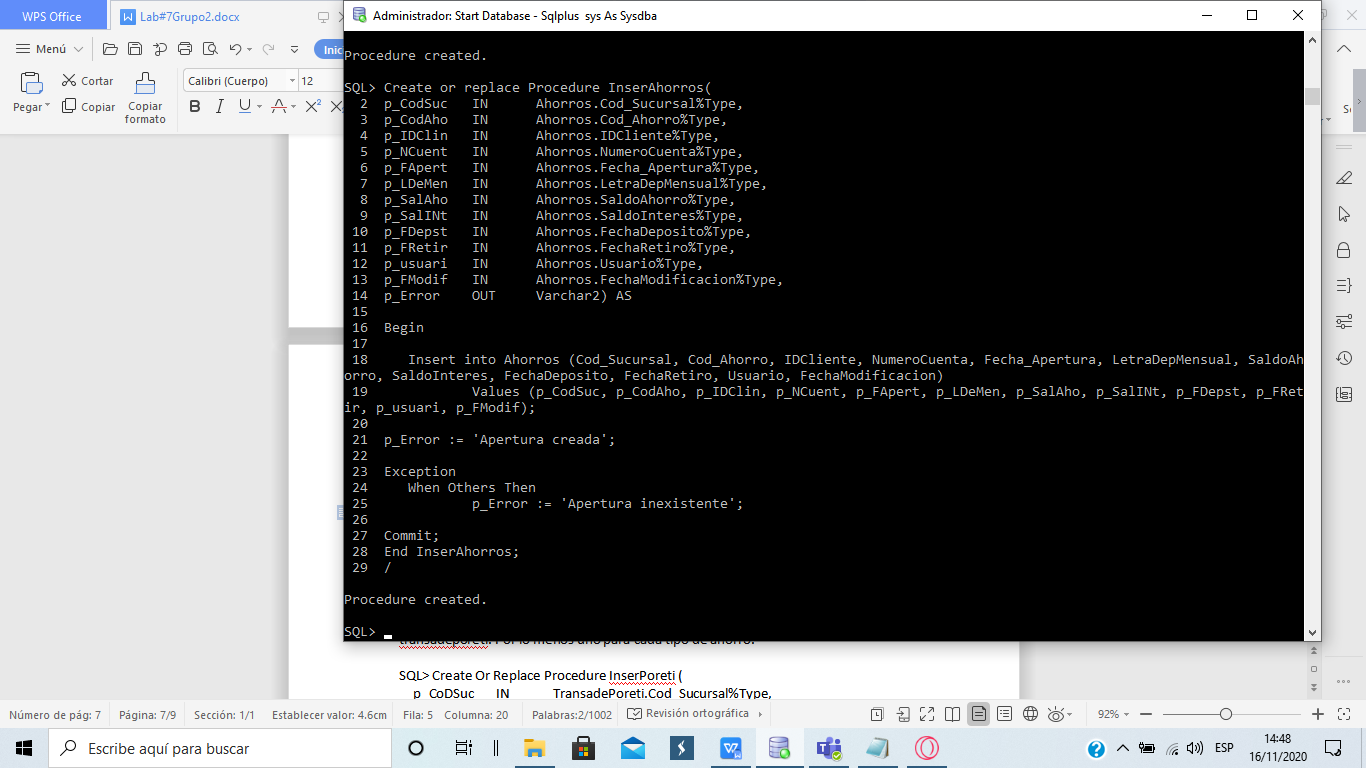
p\_Error := 'Apertura inexistente';

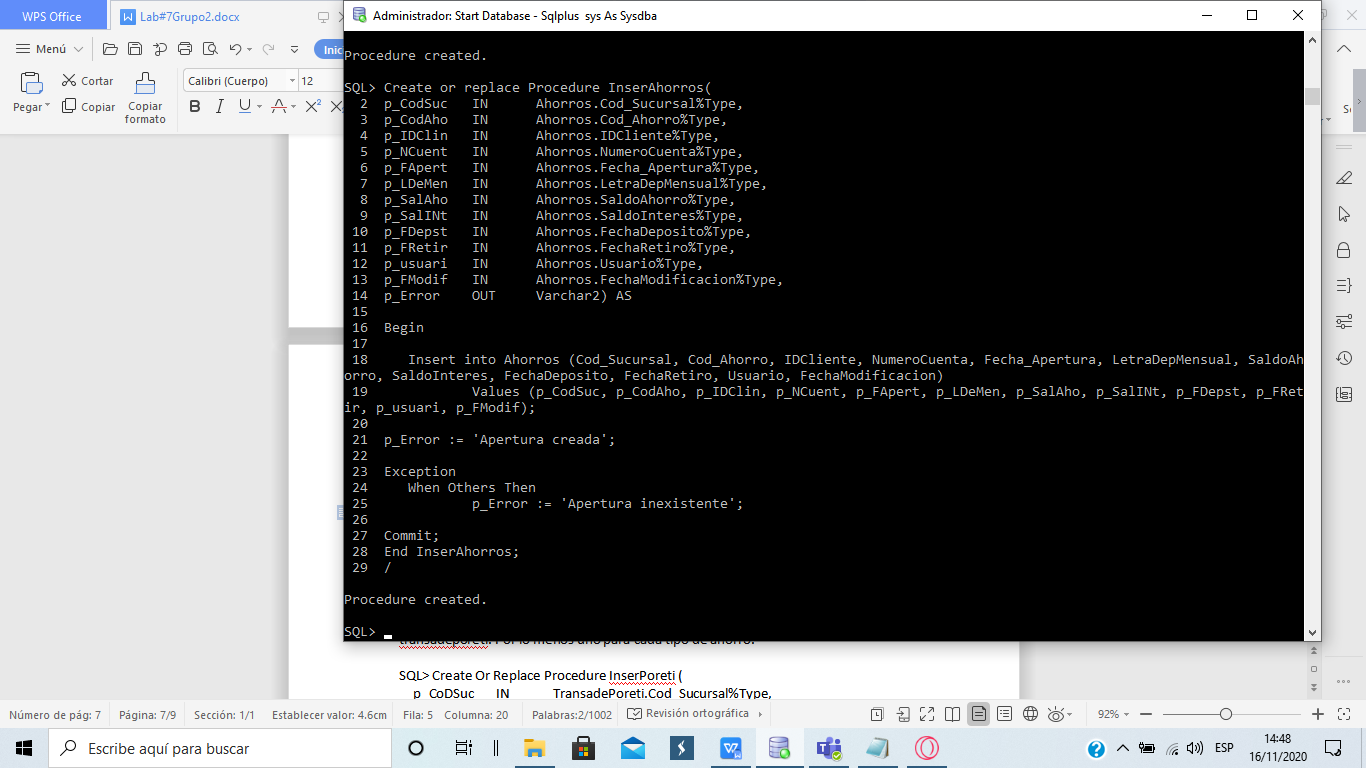
Commit;

End InserAhorros;

/

Procedure created.





**Procedimiento** almacenado para la carga o inserción de los depósitos o retiros

recibidos de los clientes para sus cuentas de ahorros que se almacena en la tabla

transadeporeti. Por lo menos uno para cada tipo de ahorro.

SQL> Create Or Replace Procedure InserPoreti (

p\_CoDSuc IN TransadePoreti.Cod\_Sucursal%Type,

p\_IDTran IN TransadePoreti.ID\_Transaccion%Type,

p\_IDClin IN TransadePoreti.IDCliente%Type,

p\_TipAho IN TransadePoreti.Tip\_Ahorro%Type,

p\_FeTran IN TransadePoreti.FechaTransaccion%Type,

p\_TiTran IN TransadePoreti.Tip\_Transac%Type,

p\_MontDR IN TransadePoreti.MontoDep\_Retiro%Type,

p\_FInser IN TransadePoreti.FechaInserccion%Type,

p\_User IN TransadePoreti.Usuario%Type,

p\_Error OUT Varchar2 ) AS

Begin

Insert Into TransadePoreti (Cod\_Sucursal, ID\_Transaccion, IDCliente, Tip\_Ahorro, FechaTransaccion, Tip\_Transac, MontoDep\_Retiro, FechaInserccion, Usuario)

Values (p\_CoDSuc, p\_IDTran, p\_IDClin, p\_TipAho, p\_FeTran, p\_TiTran, p\_MontDR, p\_FInser, p\_User);

p\_Error := 'Insercion Creada';

Exception

When Others Then

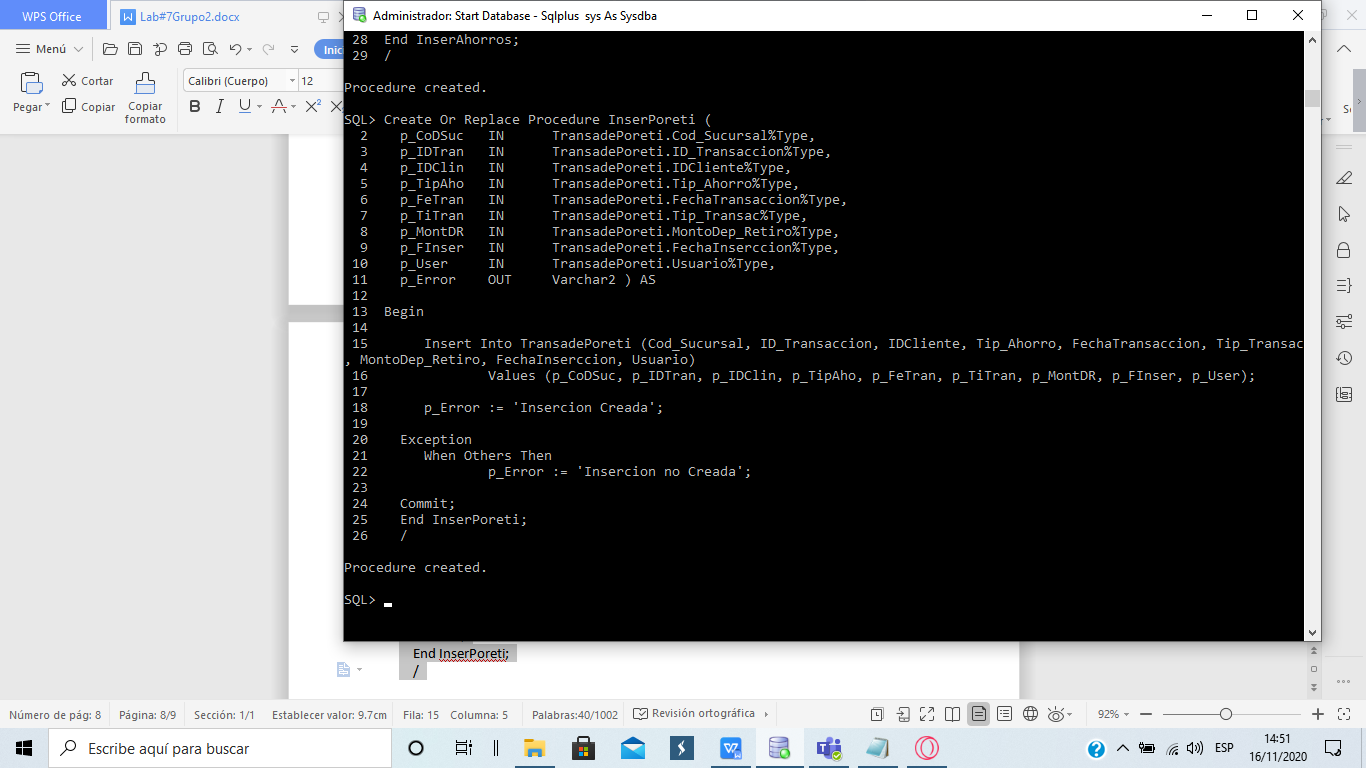
p\_Error := 'Insercion no Creada';

Commit;

End InserPoreti;

/

Procedure created.



 **Procedimiento** almacenado que actualice los depósitos o retiros de las cuentas de

ahorro correspondientes. Deberá implementar un **cursor** que busque el

depósitos/retiros insertados en la tabla uno a uno y los vaya actualizando en la

tabla de ahorros de cada cliente (proceso en lote o proceso en línea usted decide).

De la siguiente forma:

* Si el tipo de ahorro es navidad o escolar por cada deposito realizado debe calcular el interés que corresponde montodeposito \* tasade interes% que calculo que lo debe realizar una función diseñada previamente. El procedimiento debe actualizar el saldo de ahorro y el saldo interés de la cuenta de ahorro de los clientes
* Si el tipo de ahorro es corriente simplemente se realizar la aplicación del depósito o retiro a la cuenta de ahorro del cliente correspondiente. De las únicas cuentas que se puede realizar retiros es de la cuenta de ahorro corriente por lo tanto el procedimiento debe controlar esta situación.

SQL> CREATE OR REPLACE FUNCTION Calc\_Interes(

p\_monto Transadeporeti.MontoDep\_Retiro%TYPE,

p\_tasaI TiposDeAhorros.Tasa\_Interes%TYPE

)Return

NUMBER AS

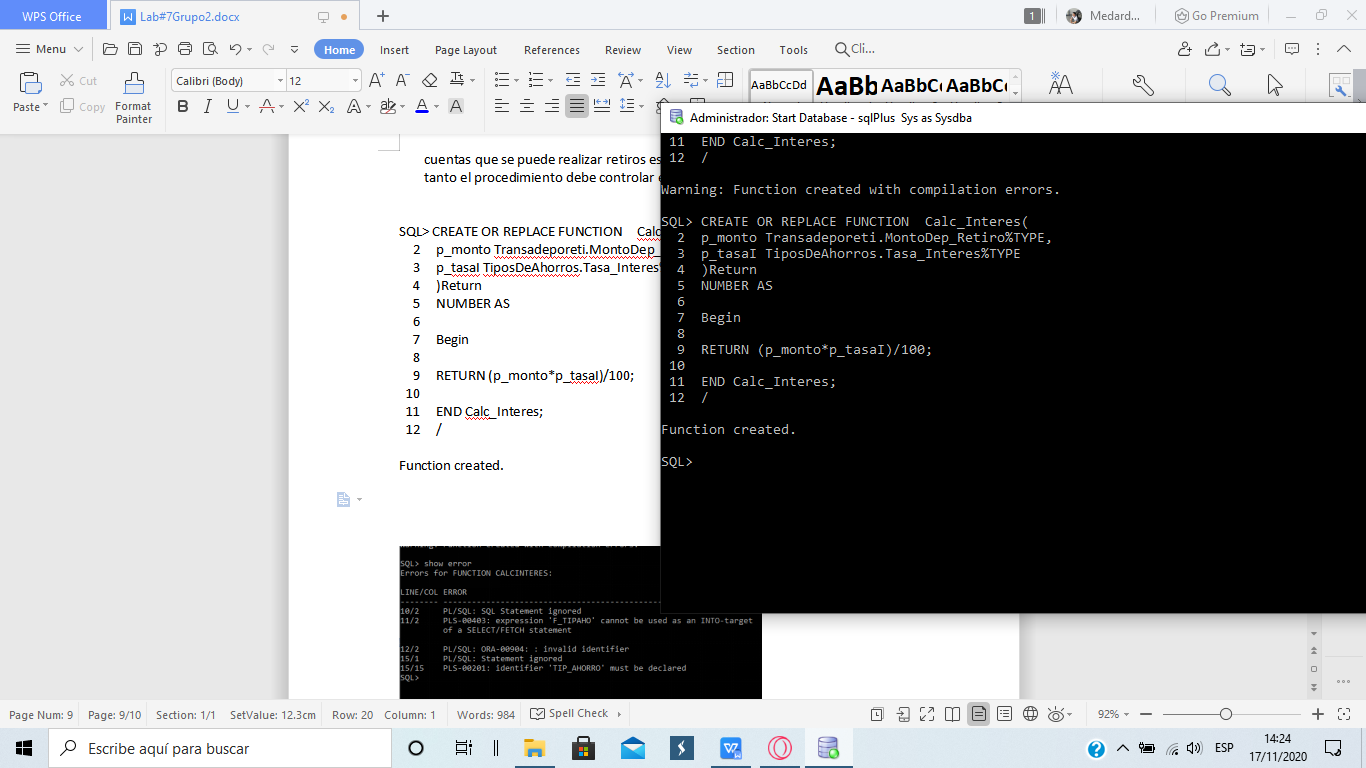
Begin

RETURN (p\_monto\*p\_tasaI)/100;

END Calc\_Interes;

/

Function created.



SQL> CREATE OR REPLACE PROCEDURE Actualizar\_DR (

p\_error out varchar2)AS

CURSOR c\_Deporeti IS

SELECT Cod\_Sucursal, IDCliente, Tip\_Ahorro, Tip\_Transac, MontoDep\_Retiro, NCuenta

From Transadeporeti;

p\_codsuc Transadeporeti.cod\_sucursal%TYPE;

p\_Ttrans Transadeporeti.ID\_Transaccion%TYPE;

p\_Idcli Transadeporeti.IDCliente%TYPE;

p\_monto Transadeporeti.MontoDep\_Retiro%TYPE;

p\_tipoaho Transadeporeti.Tip\_Ahorro%TYPE;

p\_tasaI TiposDeAhorros.Tasa\_Interes%TYPE;

p\_calcinte Ahorros.SaldoInteres%TYPE;

p\_SaldoA Ahorros.SaldoAhorro%TYPE;

p\_Ncuenta Transadeporeti.NCuenta%TYPE;

Begin

OPEN c\_Deporeti;

Loop

FETCH c\_Deporeti INTO

p\_codsuc,p\_Idcli,p\_Ttrans,p\_tipoaho,p\_monto,p\_Ncuenta;

Select Tasa\_Interes INTO p\_tasaI

From TiposDeAhorros WHERE (Cod\_Ahorro = p\_tipoaho);

SELECT SaldoAhorro INTO p\_SaldoA

FROM Ahorros

WHERE IDCliente=p\_Idcli AND cod\_sucursal = p\_codsuc AND Cod\_Ahorro = p\_tipoaho;

IF ((p\_tipoaho = 1 OR p\_tipoaho = 3) AND p\_Ttrans=1) THEN

p\_calcinte := Calc\_Interes(p\_monto,p\_tasaI);

UPDATE Ahorros

SET

SaldoAhorro = SaldoAhorro + p\_monto + p\_calcinte,

SaldoInteres = SaldoInteres + p\_calcinte,

Usuario=user,

FechaModificacion = sysdate,

FechaDeposito = sysdate

WHERE p\_Idcli = IDCliente AND p\_Ncuenta = NumeroCuenta;

ELSIF ((p\_tipoaho = 1 OR p\_tipoaho = 3)

AND p\_Ttrans=2) THEN

DBMS\_OUTPUT.PUT\_LINE('NO SE PUEDE RETIRAR DE ESTE TIPO DE CUENTA');

ELSIF (p\_tipoaho = 2 AND p\_Ttrans=1) THEN

UPDATE Ahorros

Set

SaldoAhorro = SaldoAhorro + p\_monto,

Usuario=user,

FechaModificacion = sysdate,

FechaDeposito = sysdate

WHERE p\_Idcli = IDCliente AND p\_Ncuenta = Numerocuenta;

ELSIF (p\_tipoaho = 2 AND p\_Ttrans=2) THEN

IF(p\_monto <= p\_SaldoA) THEN

UPDATE Ahorros

SET

Saldoahorro =Saldoahorro - p\_monto,

Usuario = user,

FechaModificacion = sysdate,

FechaDeposito = sysdate

WHERE p\_Idcli=IDCliente AND p\_Ncuenta=NumeroCuenta;

ELSE

DBMS\_OUTPUT.PUT\_LINE('Fondos insuficiente');

End If;

End If;

EXIT WHEN c\_Deporeti%notfound;

END LOOP;

CLOSE c\_Deporeti;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

p\_error := 'No se encontraron datos';

WHEN OTHERS THEN

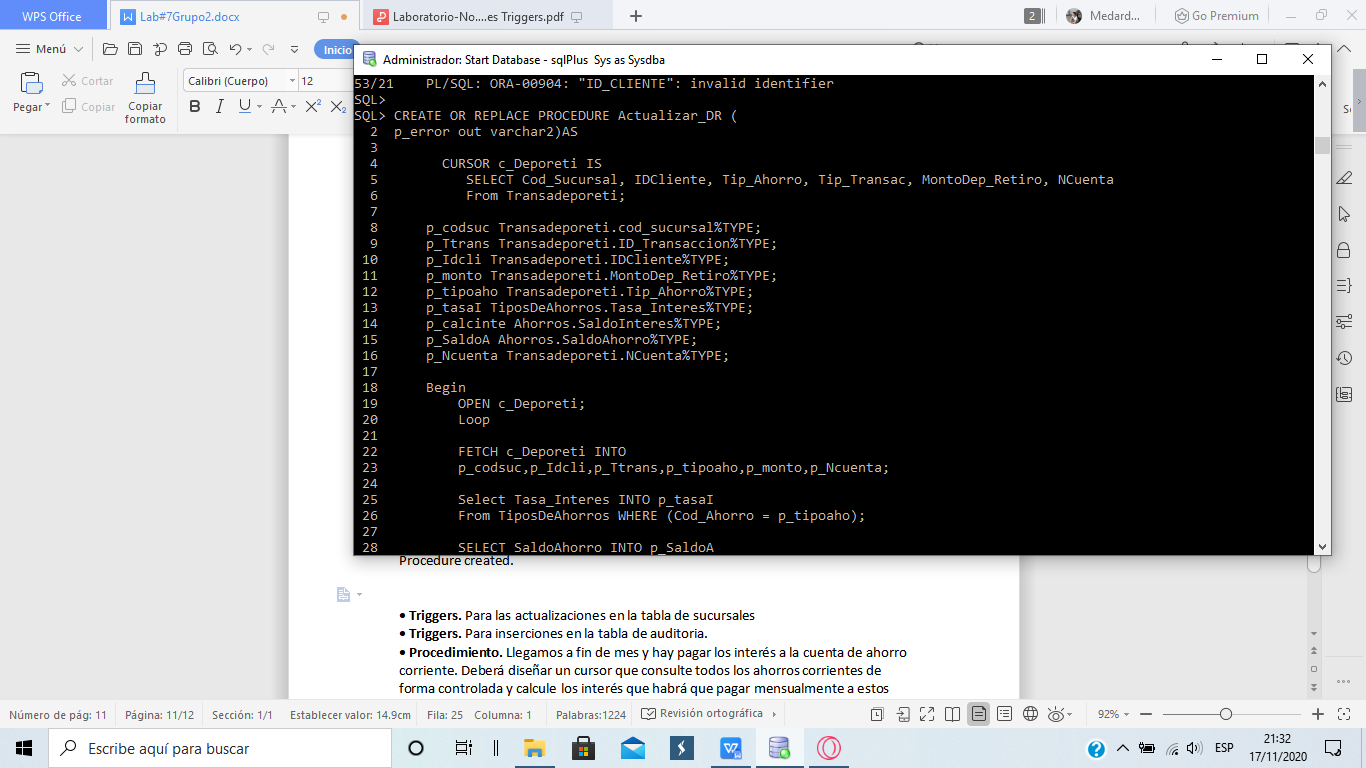
p\_error := 'No se actualizo la tabla ahorros';

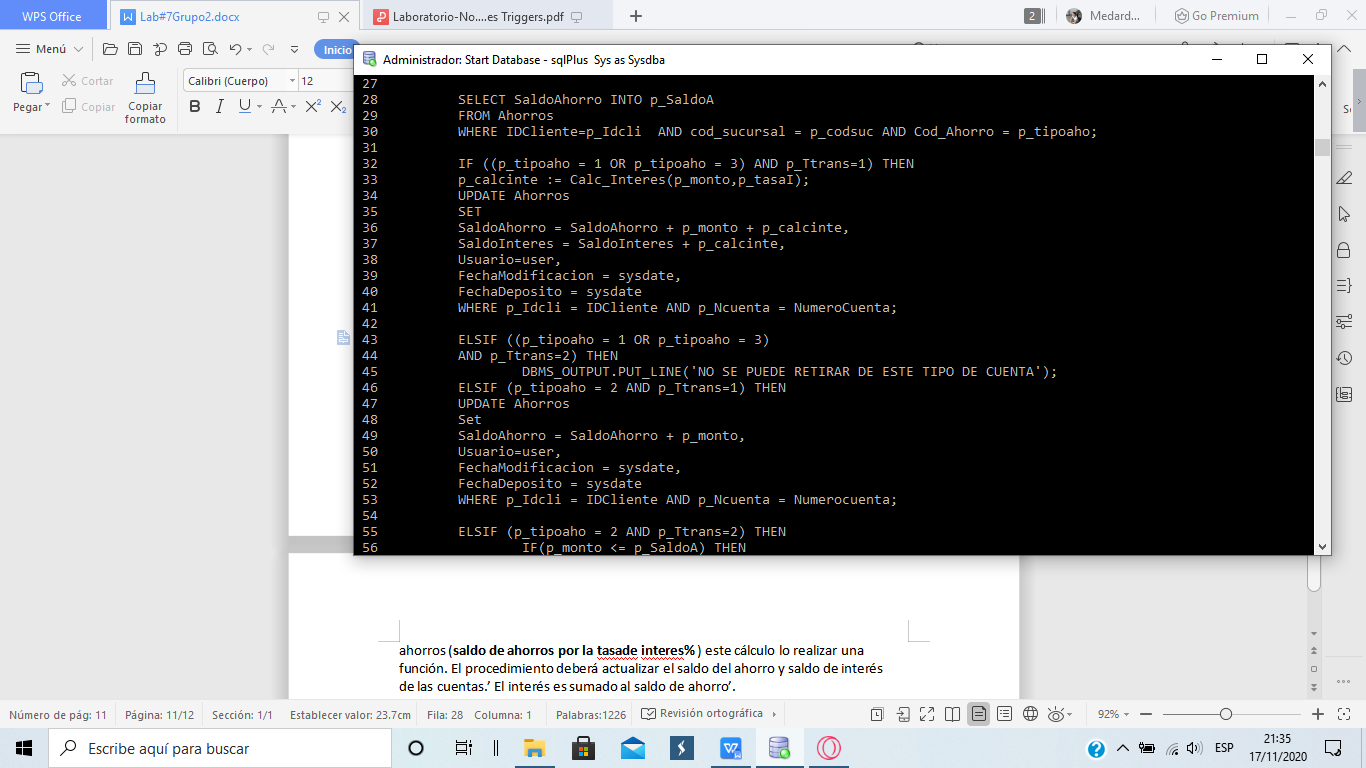
Commit;

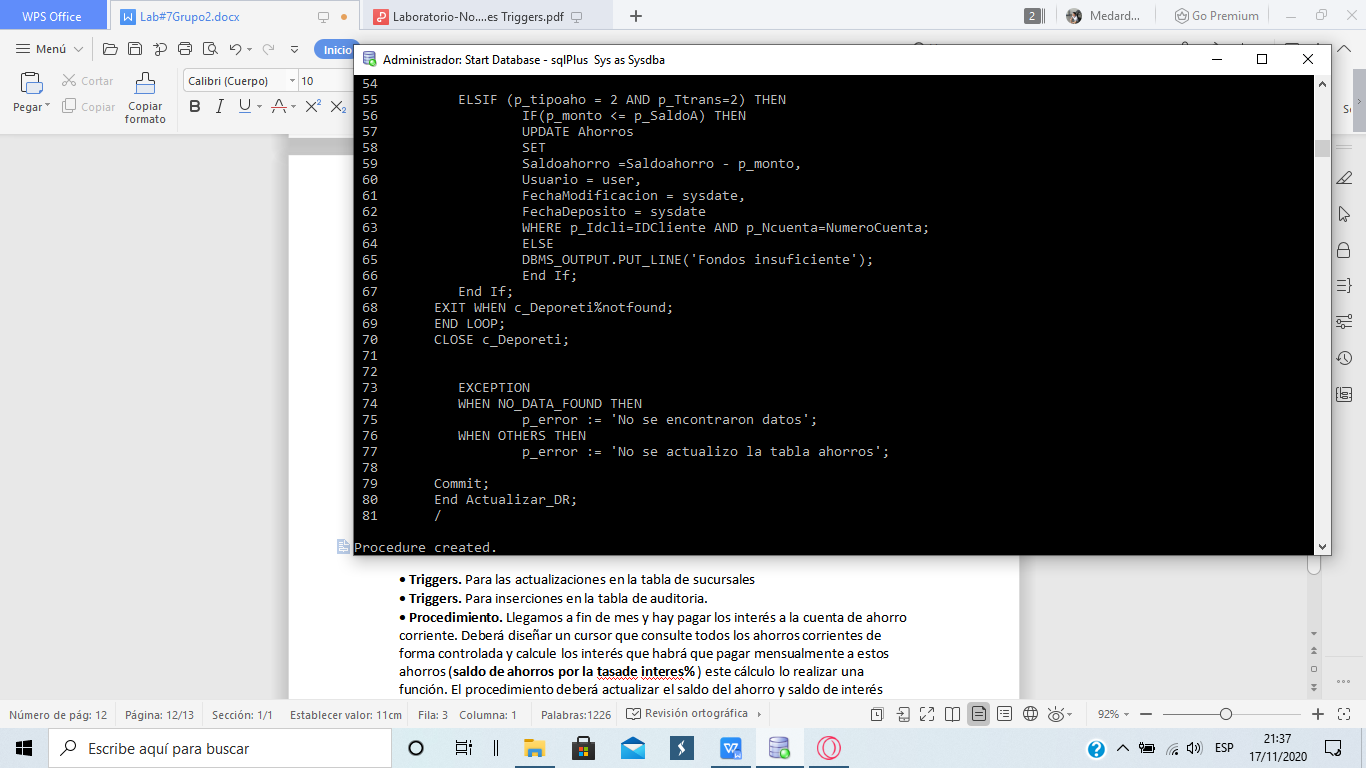
End Actualizar\_DR;

/

Procedure created.







 **Triggers.** Para las actualizaciones en la tabla de sucursales

 **Triggers.** Para inserciones en la tabla de auditoria.

 **Procedimiento.** Llegamos a fin de mes y hay pagar los interés a la cuenta de ahorro

corriente. Deberá diseñar un cursor que consulte todos los ahorros corrientes de

forma controlada y calcule los interés que habrá que pagar mensualmente a estos

ahorros (**saldo de ahorros por la tasade interes%** ) este cálculo lo realizar una

función. El procedimiento deberá actualizar el saldo del ahorro y saldo de interés

de las cuentas.’ El interés es sumado al saldo de ahorro’.

Evidencias