**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 #6

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**

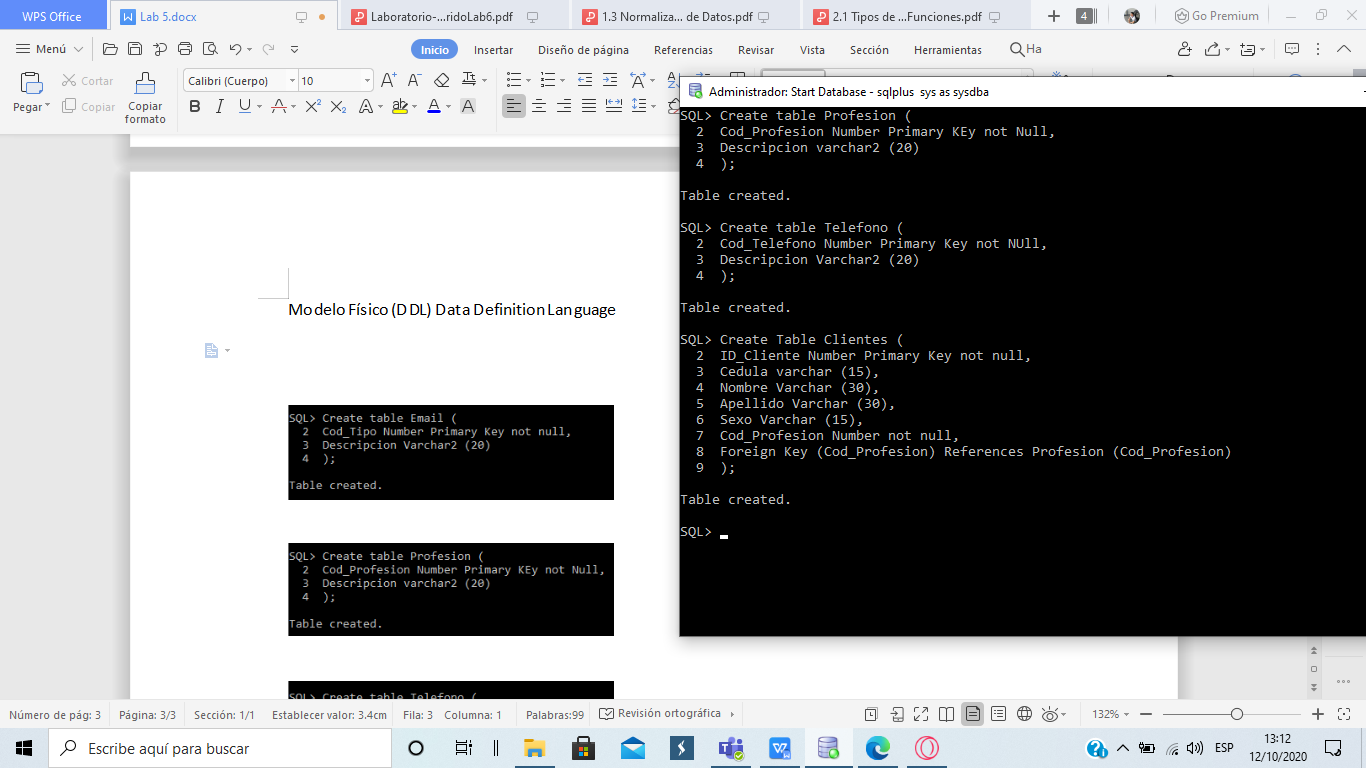
**FACULTAD DE INGENIERIA DE SISTEMAS COMPUTACIONES**

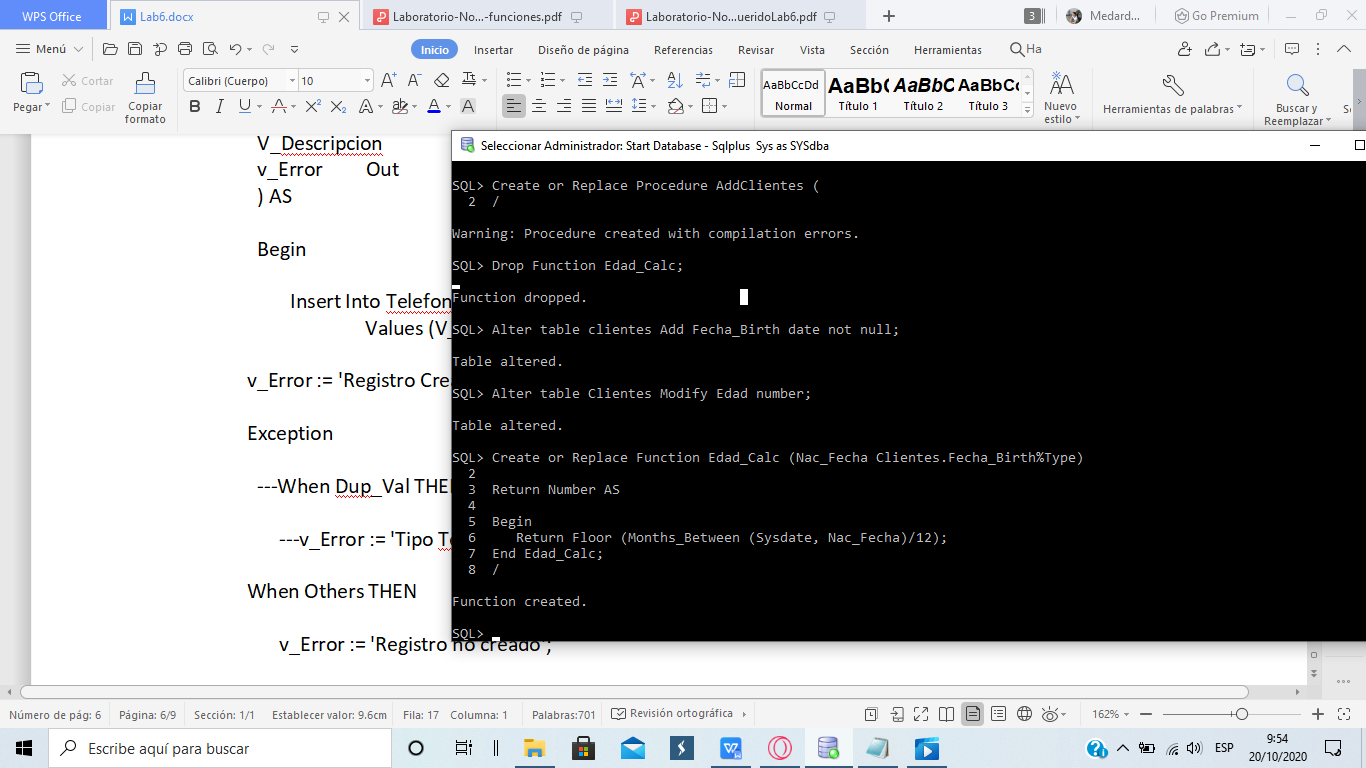
**LICENCIATURA EN INGENIERIA DE SISTEMAS DE INFORMACION**

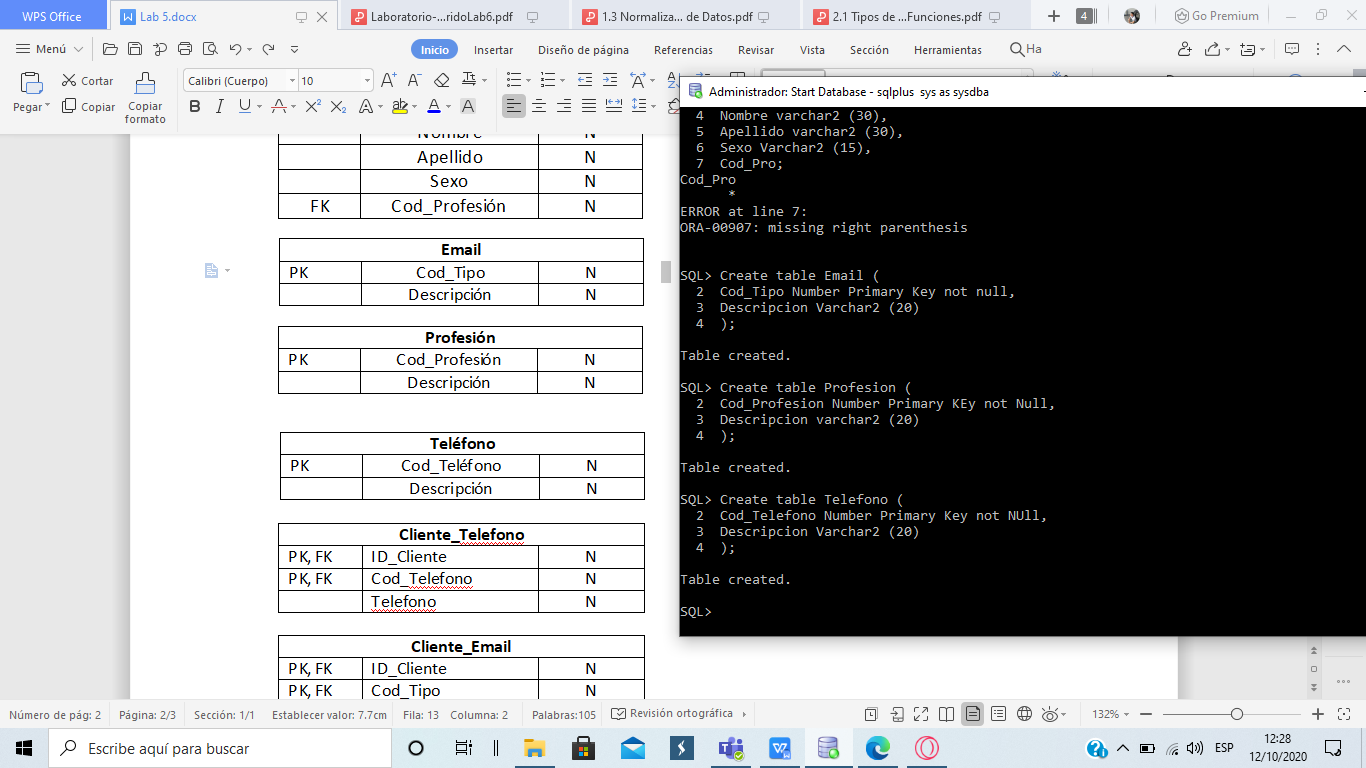
**SISTEMAS DE BASE DE DATOS II**

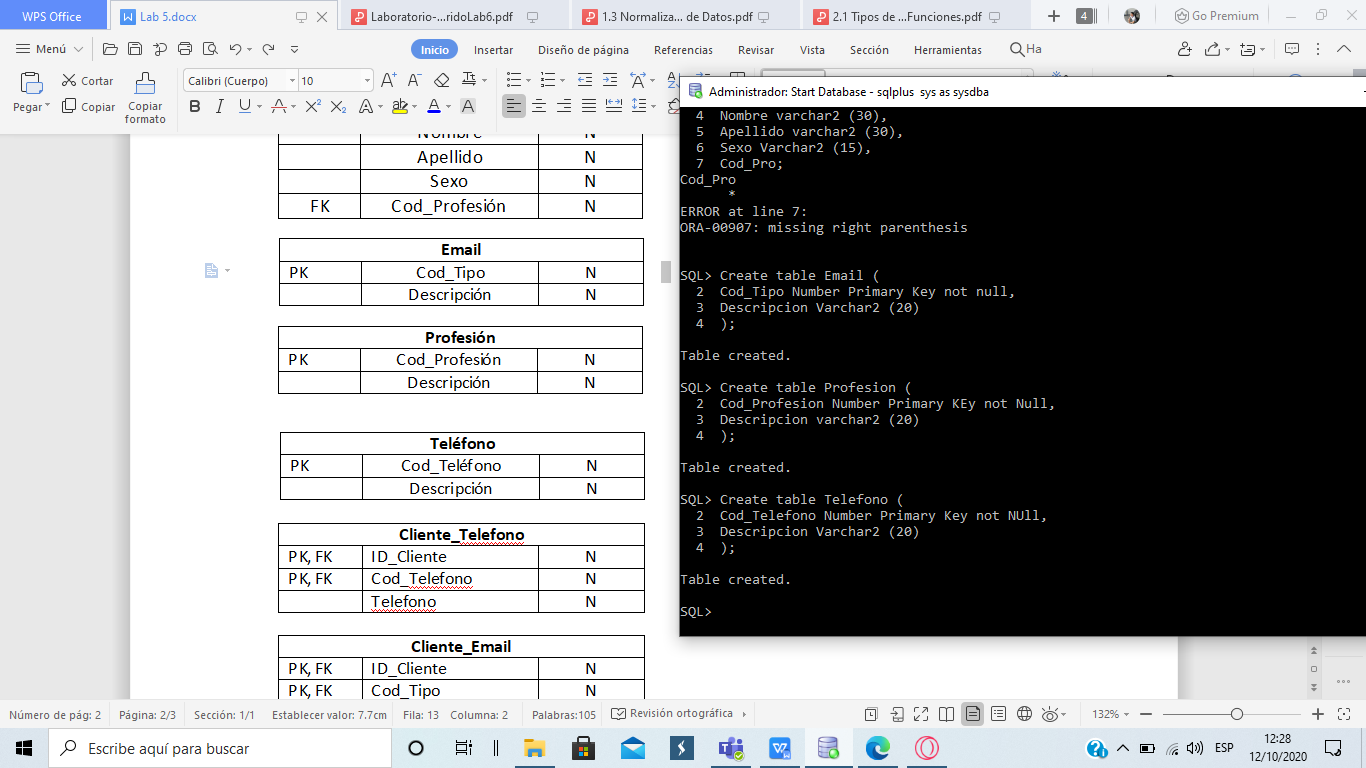
**LABORATORIO NO.6**

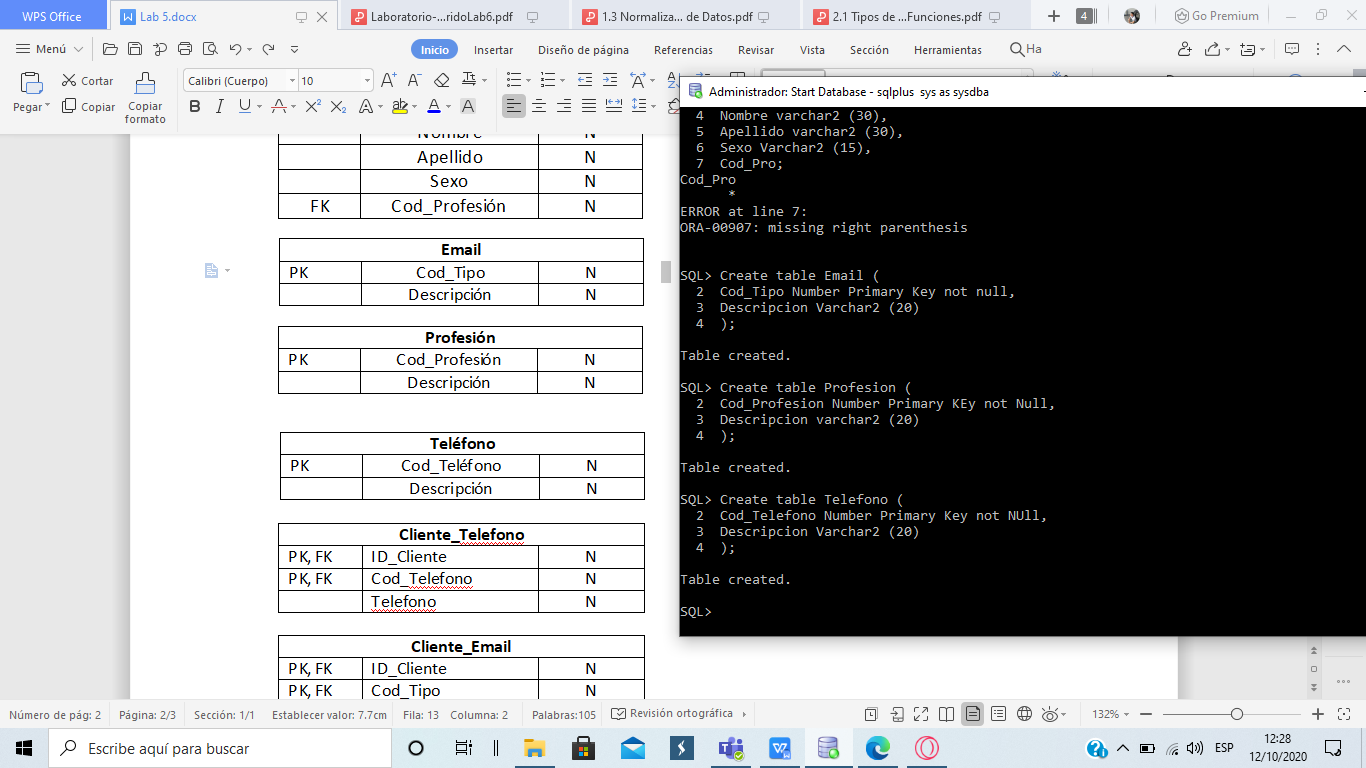
Modelo Físico (DDL) Data Definition Language

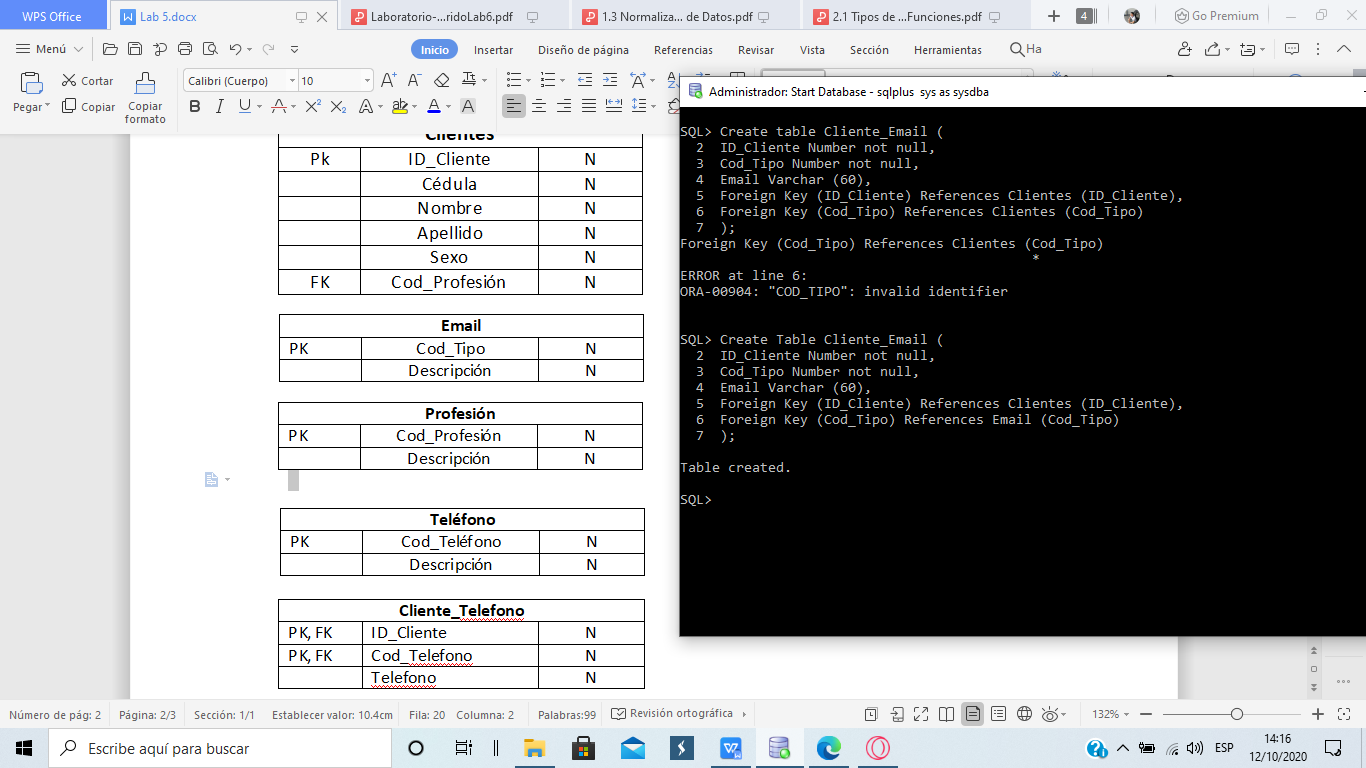


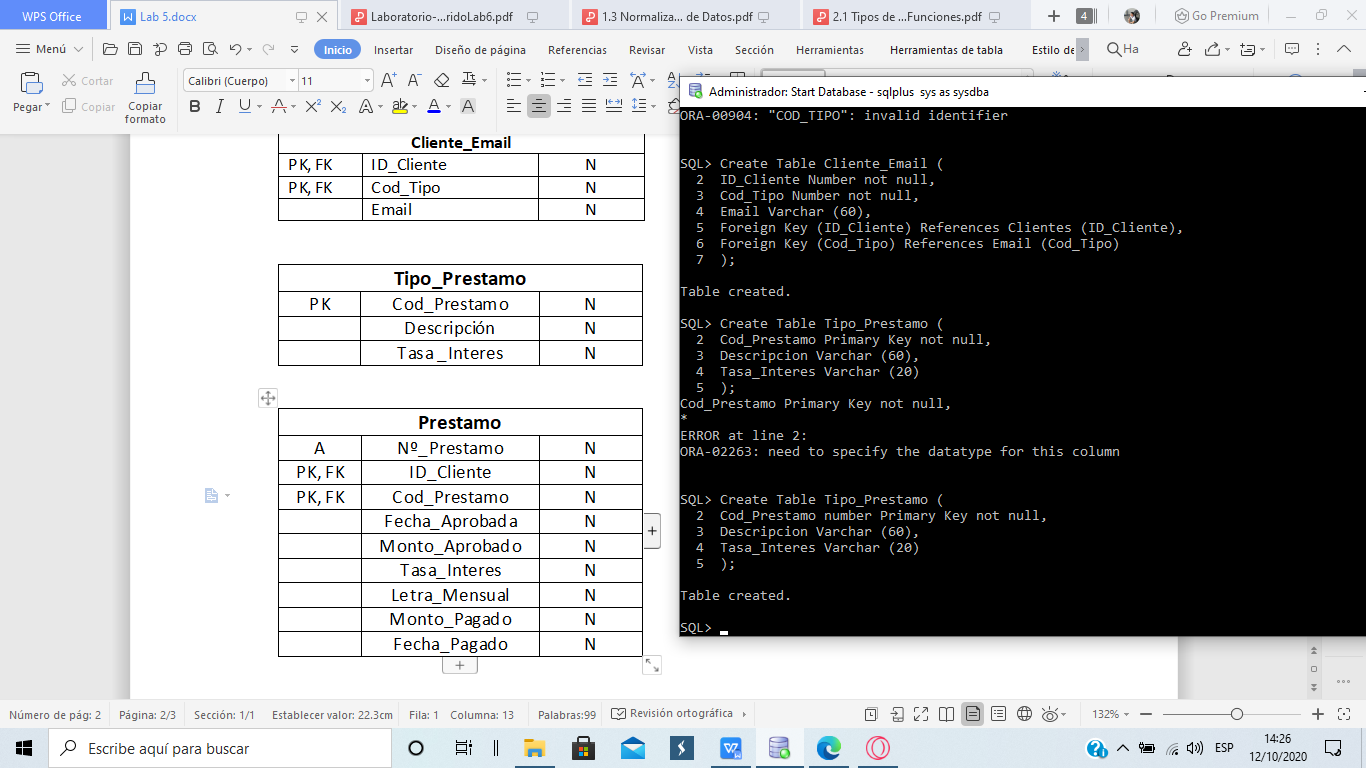


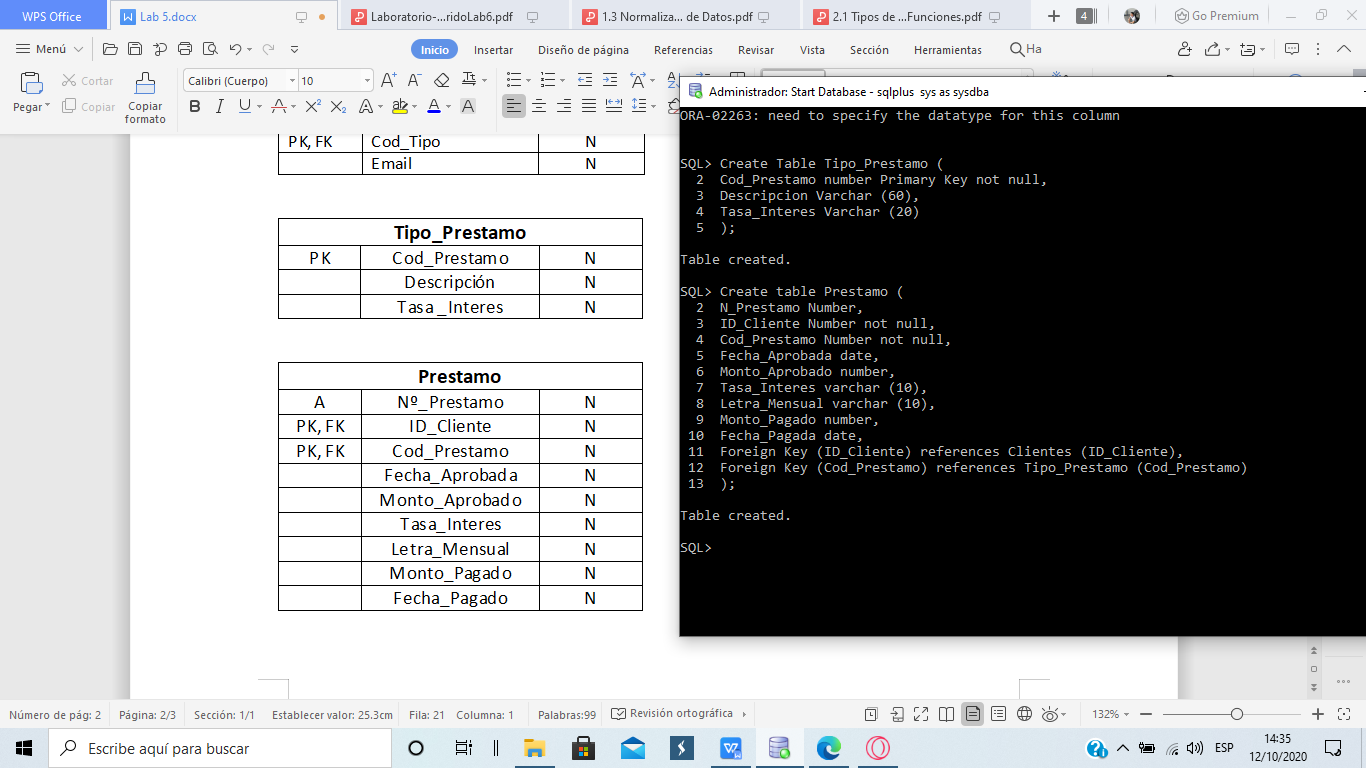


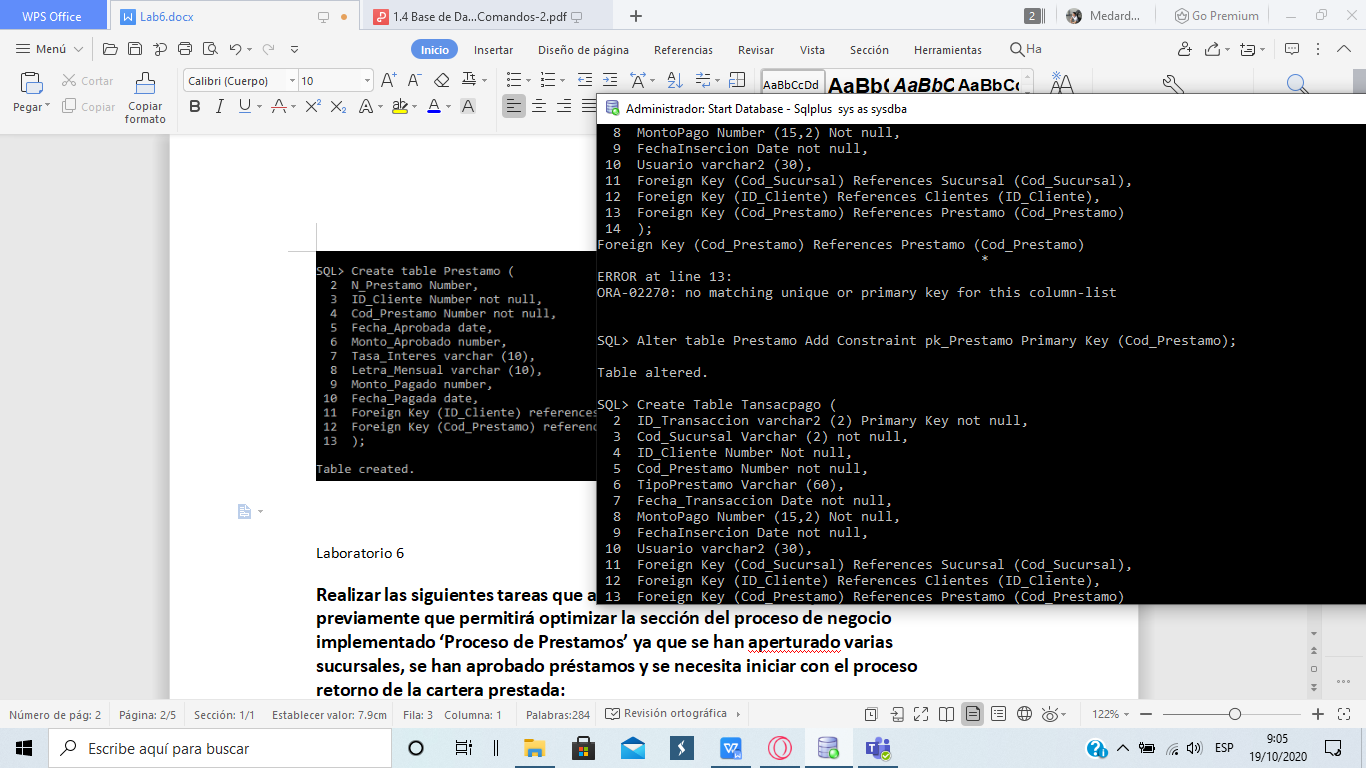












Laboratorio 6

**Realizar las siguientes tareas que afectan el Modelo Físico implementado**

**previamente que permitirá optimizar la sección del proceso de negocio**

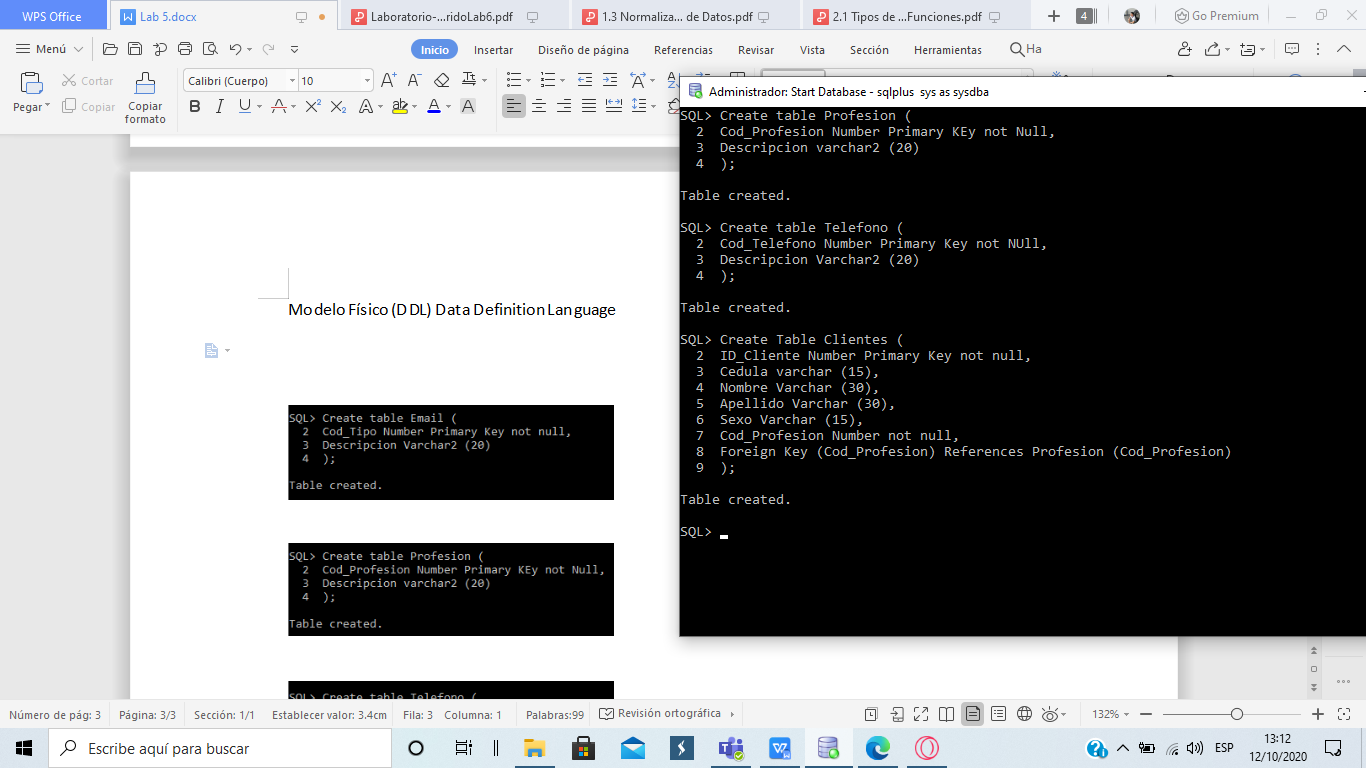
**implementado ‘Proceso de Prestamos’ ya que se han aperturado varias**

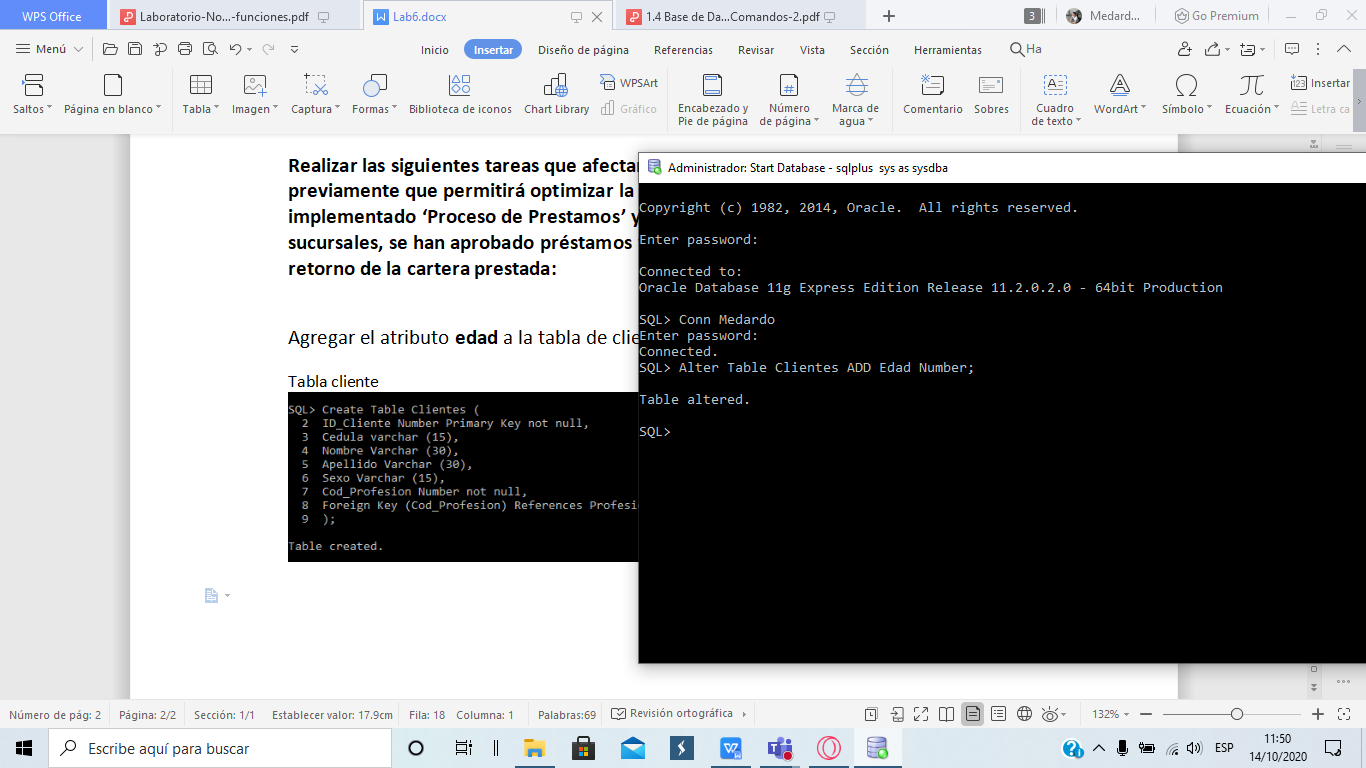
**sucursales, se han aprobado préstamos y se necesita iniciar con el proceso**

**retorno de la cartera prestada:**

Agregar el atributo **edad** a la tabla de clientes proporcionada.

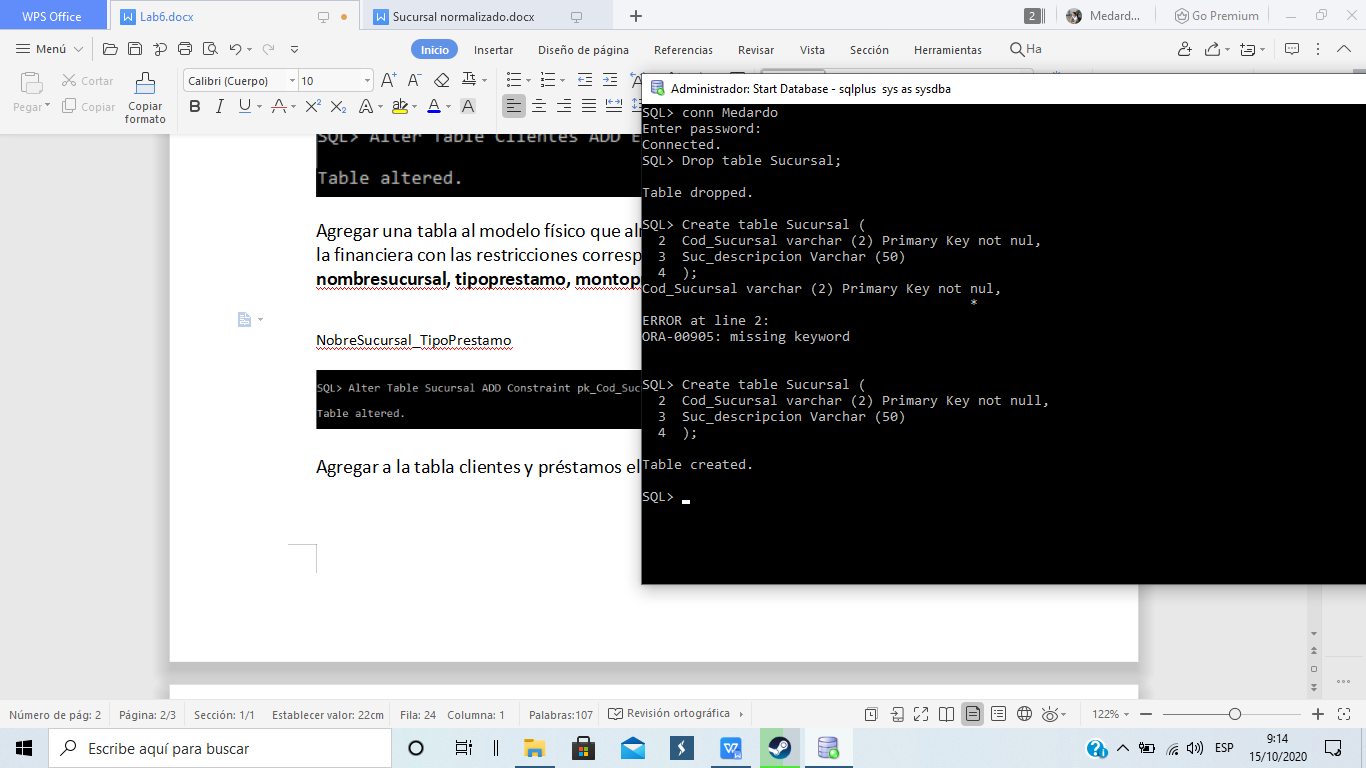
Tabla cliente

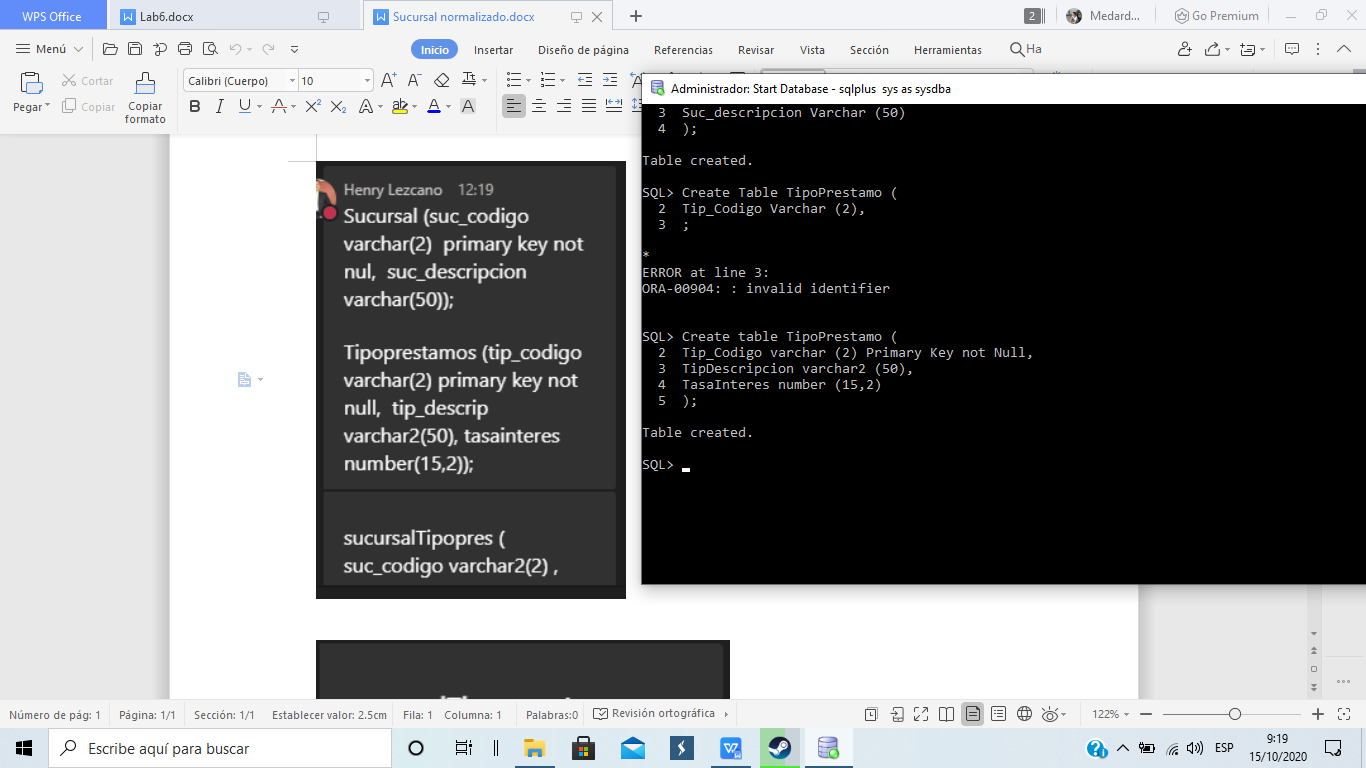


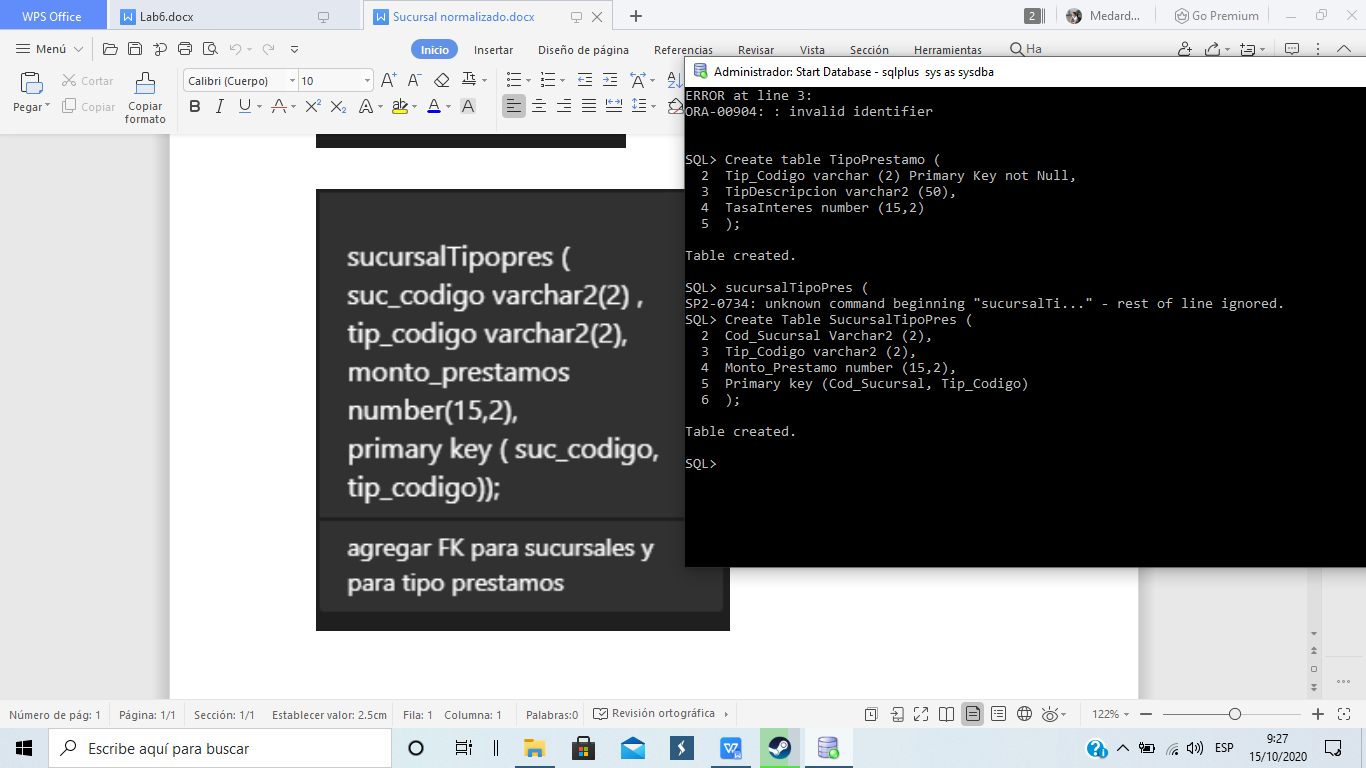


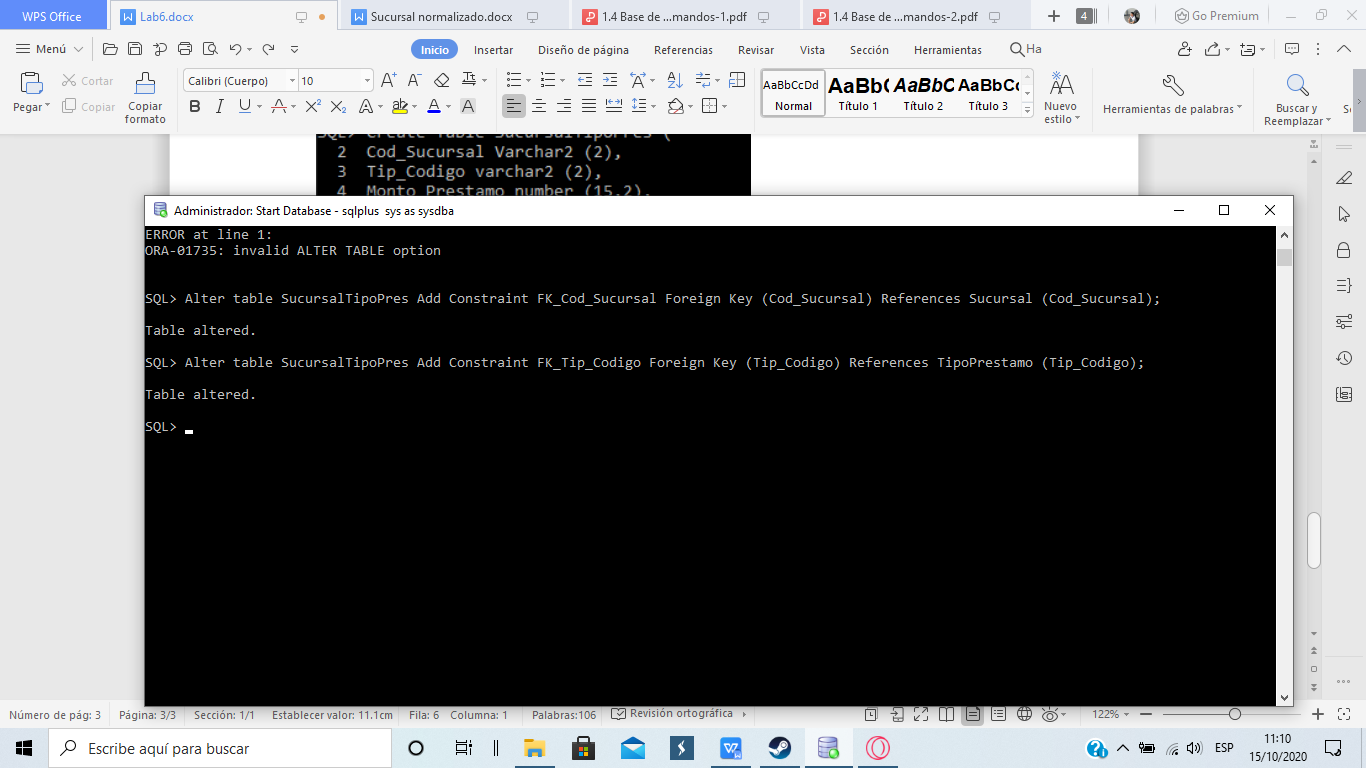
Agregar una tabla al modelo físico que almacenes las sucursales de la empresa

la financiera con las restricciones correspondientes: **Sucursal (cod\_surcursal, nombresucursal, tipoprestamo, montoprestamos)**

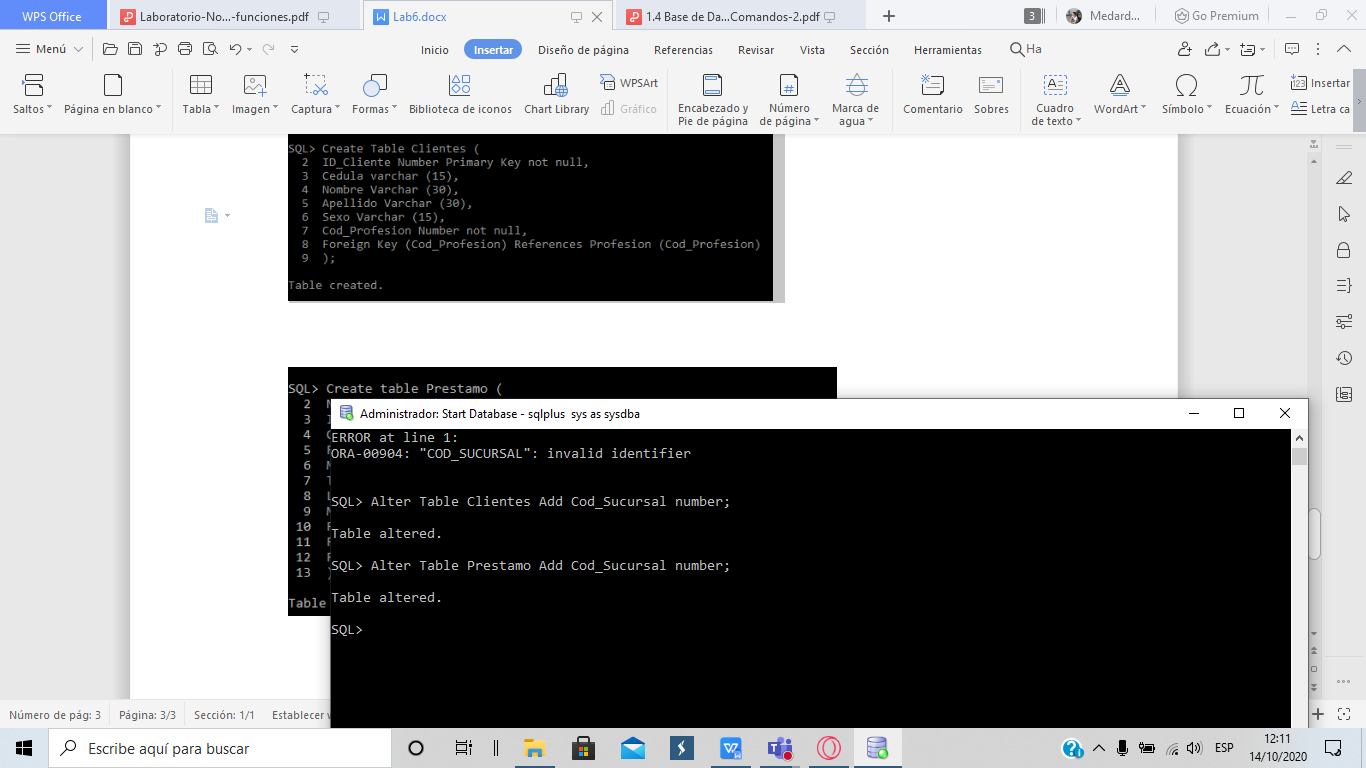


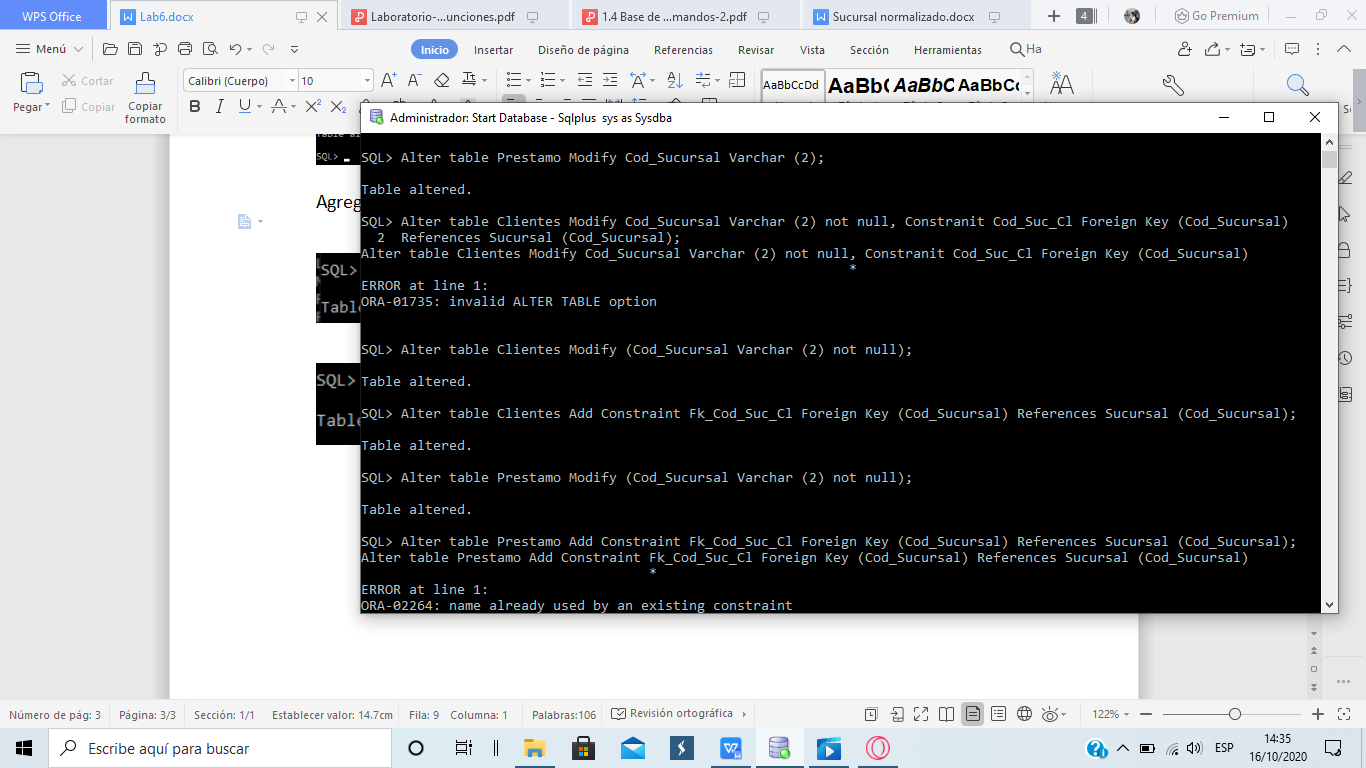


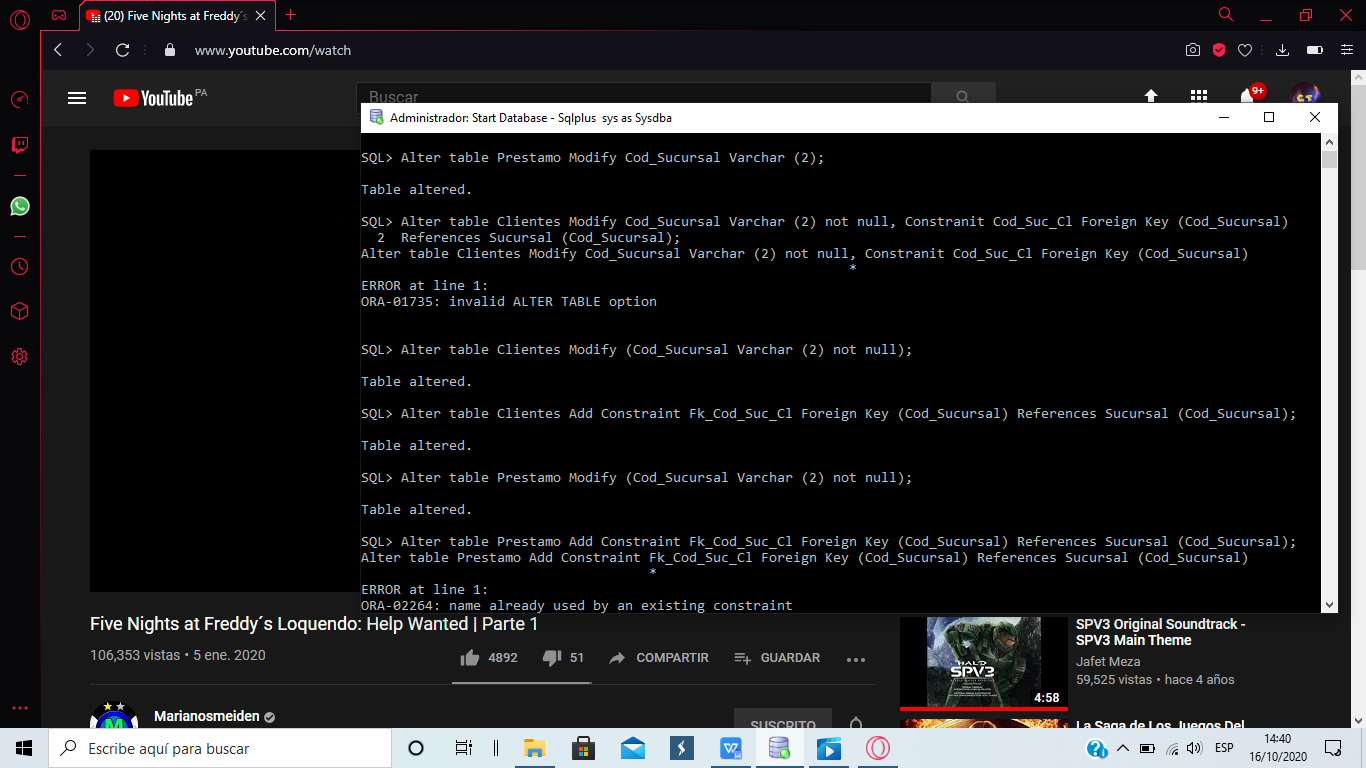


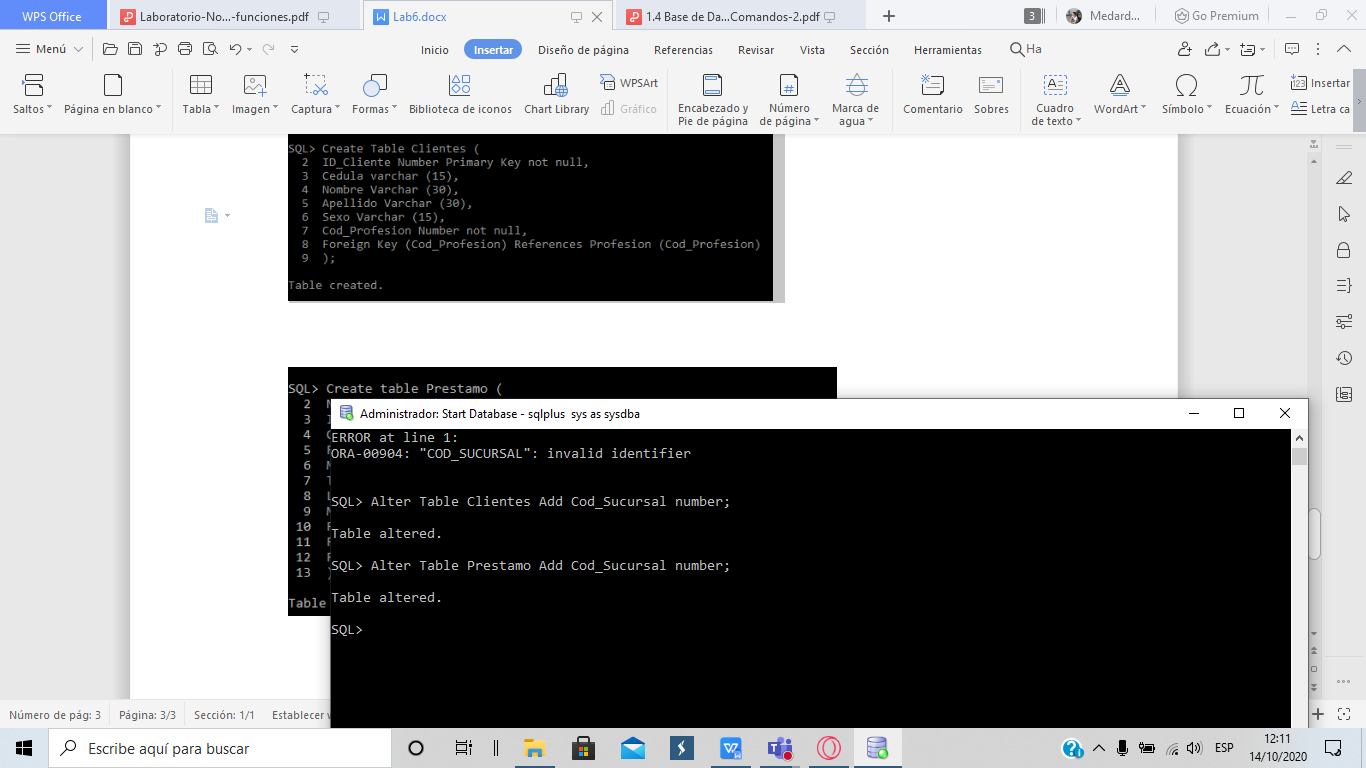


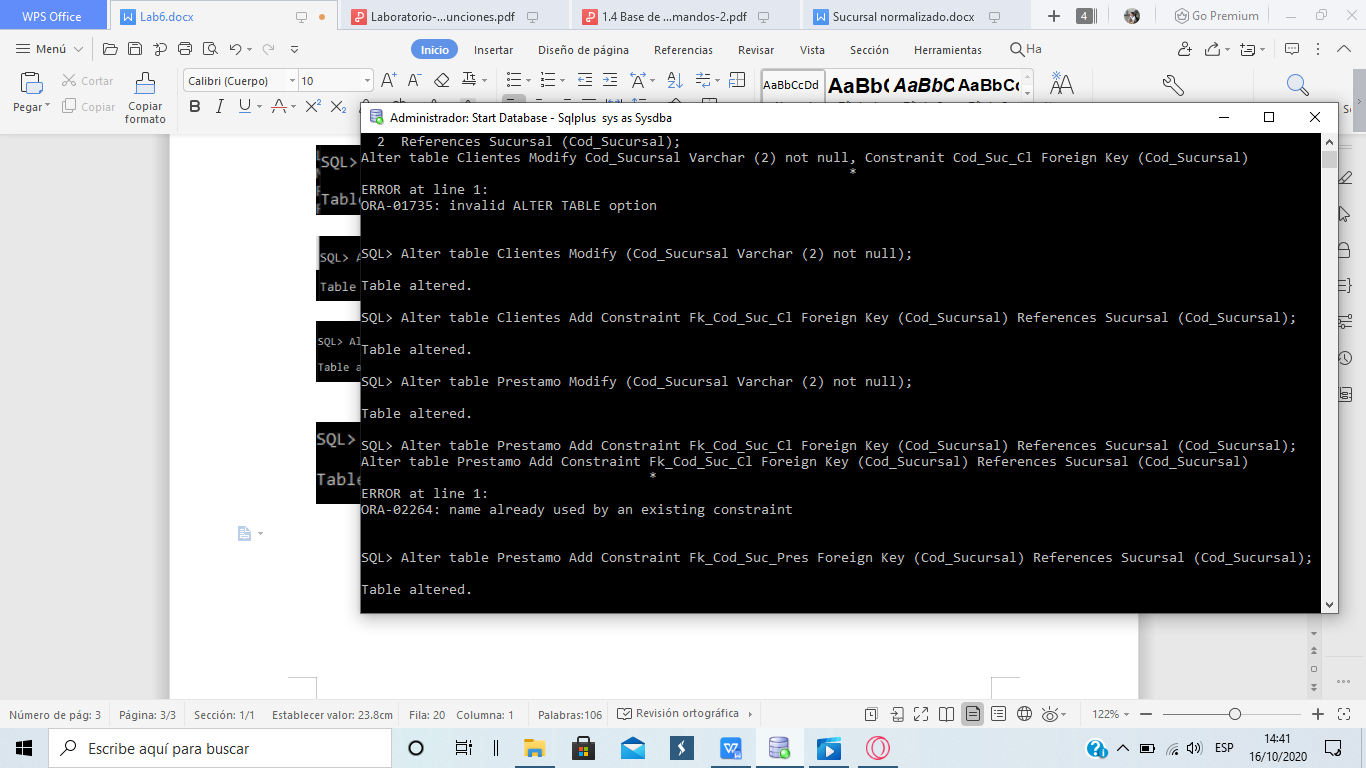
Agregar a la tabla clientes y préstamos el atributo de **código de sucursal**.

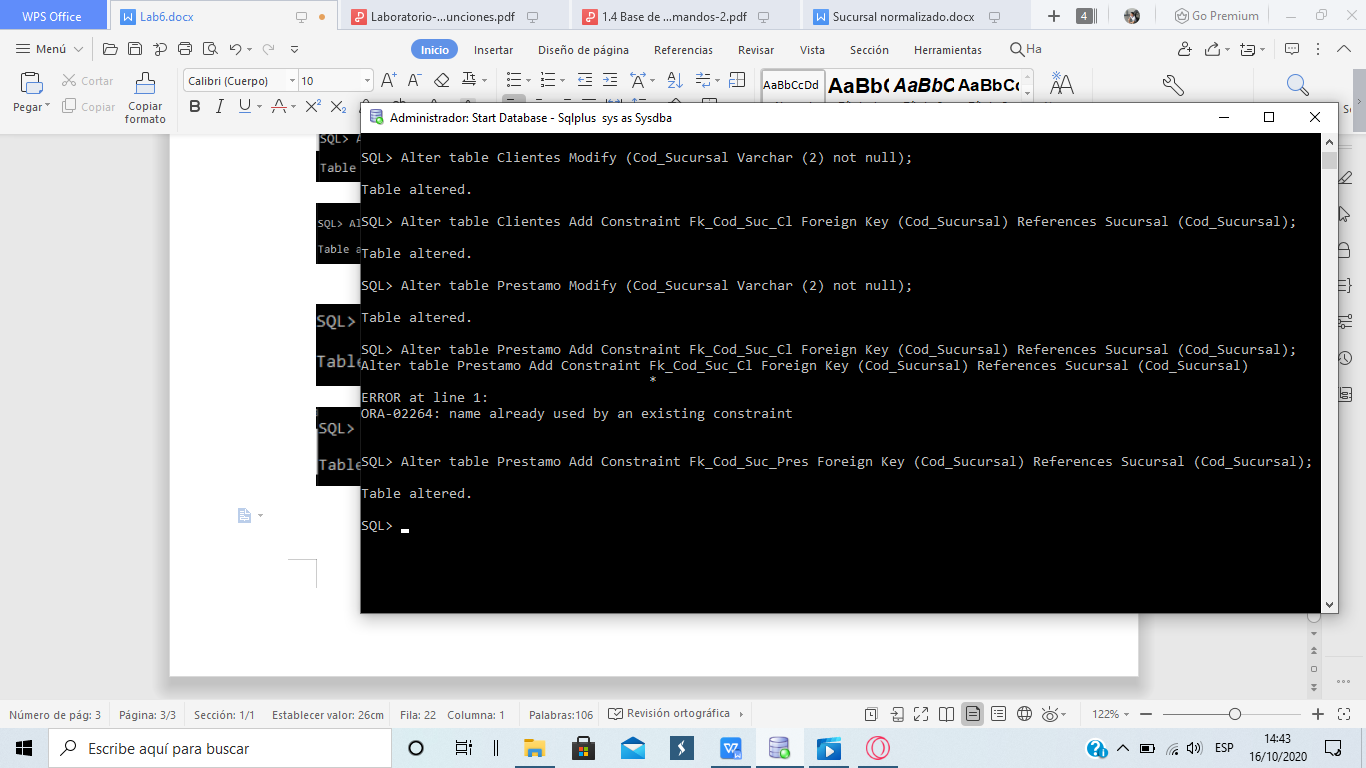










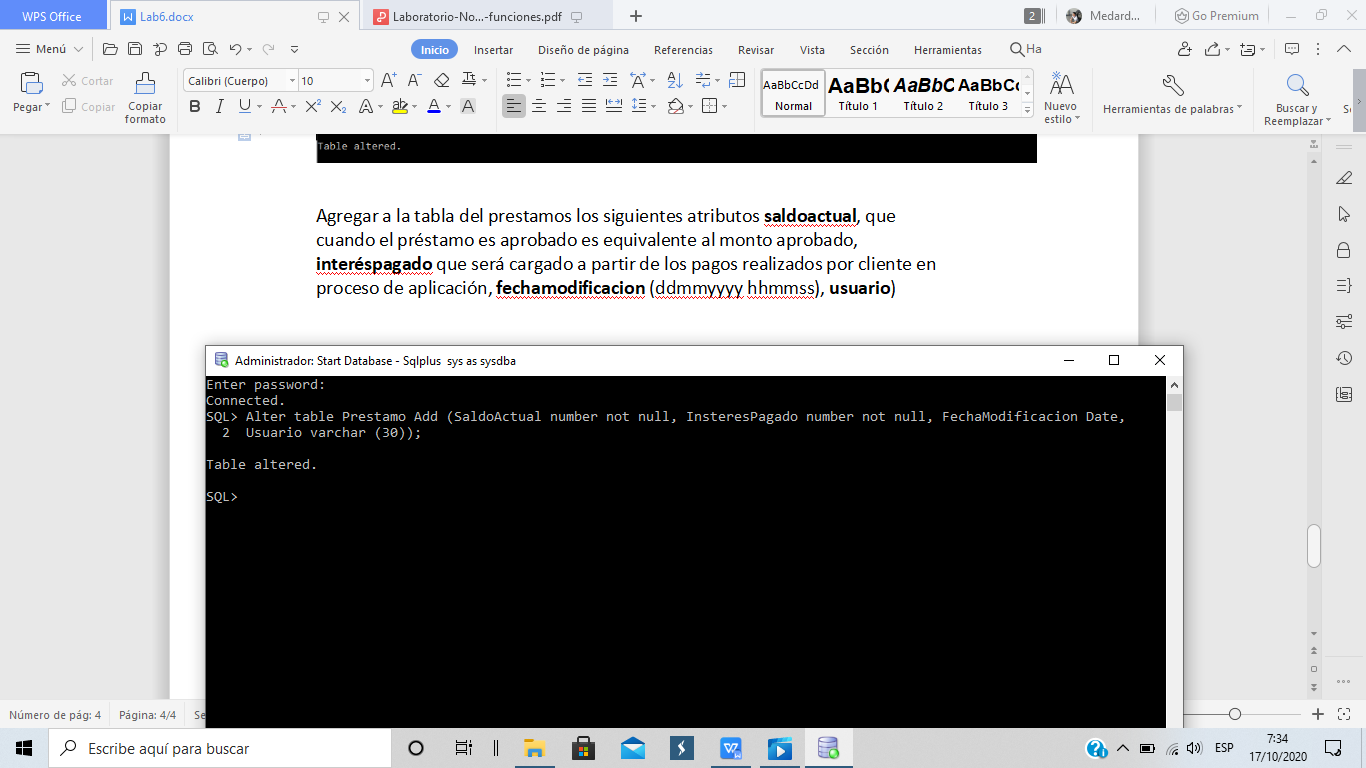


Agregar a la tabla del prestamos los siguientes atributos **saldoactual**, que

cuando el préstamo es aprobado es equivalente al monto aprobado,

**interéspagado** que será cargado a partir de los pagos realizados por cliente en

proceso de aplicación, **fechamodificacion** (ddmmyyyy hhmmss), **usuario**)



Agregar una tabla transaccional para recibir los pagos de los clientes con las

restricciones correspondiente que contendrá la siguiente atributos:

**Transacpagos (cod\_sucursal, id\_transacción, id\_cliente, tipoprestamo,**

**fechatransaccion, monto del pago, fechainsercion(dd/mm/yyyy,**

**hh:mm:ss, usuario)**

Modelo Lógico Normalizado

|  |  |  |
| --- | --- | --- |
| **Tansacpagos** | | |
| PK | ID\_Transacción | N |
| Fk | Cod\_Sucursal | N |
| Fk | ID\_Cliente | N |
| Fk | Cod\_Prestamo | N |
|  | TipoPrestamo | N |
|  | Fecha\_Transaccion | N |
|  | MontoPago | N |
|  | FechaInsercion | N |
|  | Usuario | N |

SQL> Create Table Tansacpago (

ID\_Transaccion varchar2 (2) Primary Key not null,

Cod\_Sucursal Varchar (2) not null,

ID\_Cliente Number Not null,

Cod\_Prestamo Number not null,

TipoPrestamo Varchar (60),

Fecha\_Transaccion Date not null,

MontoPago Number (15,2) Not null,

FechaInsercion Date not null,

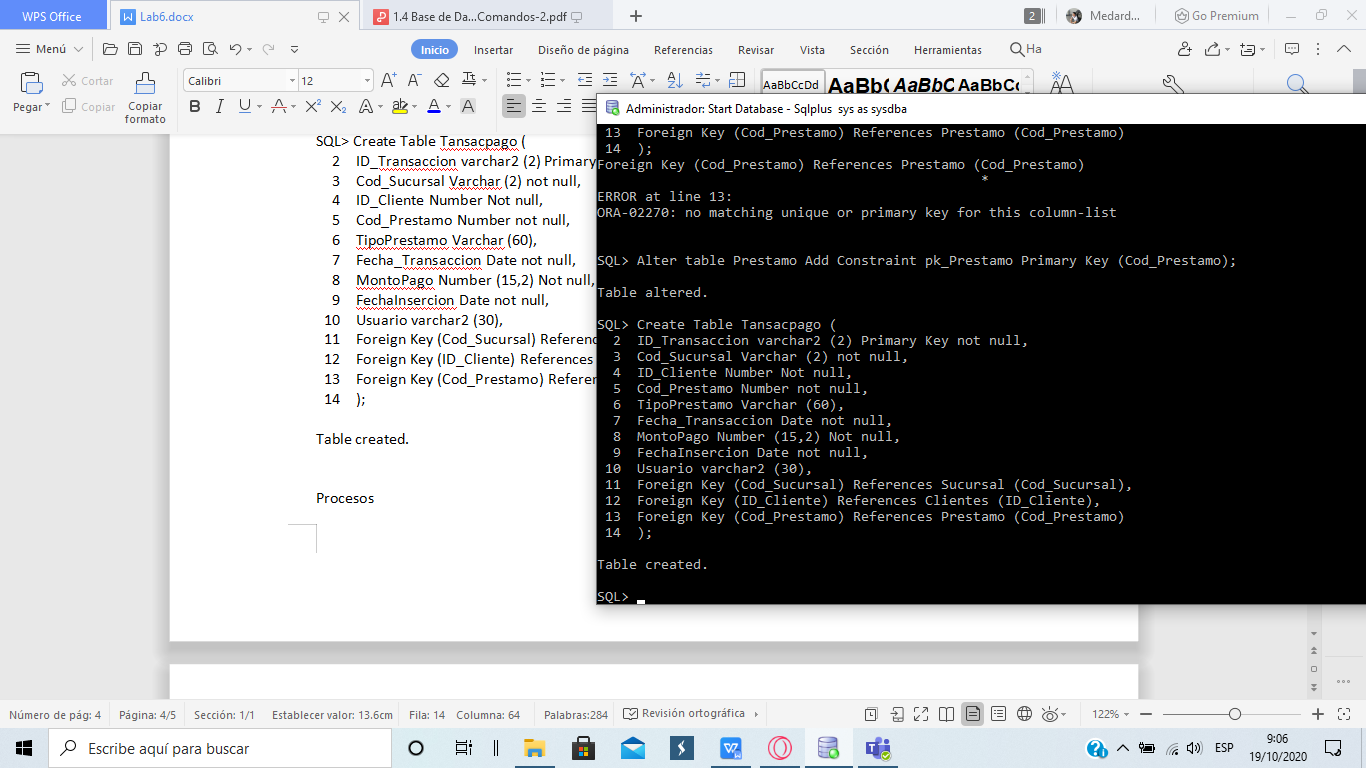
Usuario varchar2 (30),

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

Foreign Key (ID\_Cliente) References Clientes (ID\_Cliente),

Foreign Key (Cod\_Prestamo) References Prestamo (Cod\_Prestamo)

);



Procesos

Procedimiento almacenado para la carga o inserción de las tablas paramétricas

* Tipo de Teléfonos
* Tipo de Correos
* Profesiones,
* Sucursales,
* Tipos de prestamos

Tipos De Teléfono

SQL> Create or Replace Procedure Tipos\_Telefono (

V\_CODTelefono Telefono.Cod\_Telefono%Type,

V\_Descripcion Telefono.Descripcion%Type,

v\_Error Out Varchar2

) AS

Begin

Insert Into Telefono (Cod\_Telefono, Descripcion)

Values (V\_CODTelefono, V\_Descripcion);

v\_Error := 'Registro Creado';

Exception

---When Dup\_Val THEN

---v\_Error := 'Tipo Telefono Existente';

When Others THEN

v\_Error := 'Registro no creado';

Commit;

End Tipos\_Telefono;

/

Procedure created.

Tipos de Correos

SQL> Create or Replace Procedure Tip\_Email (

p\_TipMail Email.Cod\_Tipo%Type,

p\_Descrip Email.Descripcion%Type,

p\_Error out Varchar2

) AS

Begin

Insert INTO Email (Cod\_Tipo, Descripcion)

Values (p\_TipMail, p\_Descrip);

p\_Error := 'Correo creado';

Exception

---When Dup\_Val THEN

---p\_Error := 'Tipo Correo Existente';

When Others THEN

p\_Error := 'Correo no creado';

End Tip\_Email;

/

Procedure created.

Profesiones

SQL> Create or Replace Procedure AddProfesion (

p\_IDProfesion Profesion.Cod\_Profesion%Type,

p\_Descripcion Profesion.Descripcion%Type,

p\_Error Out Varchar2

) AS

Begin

Insert Into Profesion (Cod\_Profesion, Descripcion)

Values (p\_IDProfesion, p\_Descripcion);

p\_Error := 'Profesion Creada';

Exception

---When Dup\_Val THEN

---p\_Error := 'Profesion Existente';

When Others THEN

p\_Error := 'Profesion no creada';

Commit;

End AddProfesion;

/

Procedure created.

Sucursal

SQL> Create or replace Procedure AddSucursal (

p\_IDSucursal Sucursal.Cod\_SUcursal%Type,

p\_SucDescrip Sucursal.Suc\_Descripcion%Type,

p\_Error out Varchar2

) AS

Begin

Insert INTO Sucursal (Cod\_Sucursal, Suc\_descripcion)

Values (p\_IDSucursal, p\_SucDescrip);

p\_Error := 'Sucursal Creada';

Exception

---When Dup\_Val Then

---p\_Error := 'Sucursal Existente';

When Others Then

p\_Error := 'Sucursar no Creada';

Commit;

End AddSucursal;

/

Procedure created.

Tipos de Prestamos

SQL> Create OR Replace Procedure AddTipPrestamo (

p\_IDPrestamo Tipo\_Prestamo.Cod\_Prestamo%Type,

p\_Descripcion Tipo\_Prestamo.Descripcion%Type,

p\_TasaInteres Tipo\_Prestamo.Tasa\_Interes%Type,

p\_Error Out Varchar2

) AS

Begin

Insert Into Tipo\_Prestamo (Cod\_Prestamo, Descripcion, Tasa\_Interes)

Values (p\_IDPrestamo, p\_Descripcion, p\_TasaInteres);

p\_Error := 'Prestamo Creado';

Exception

---When Data\_Found Then

--- p\_Error := 'Prestamo Existente';

When Others Then

p\_Error := 'Prestamo no creado';

Commit;

End AddTipPrestamo;

/

Procedure created.

**Procedimiento** almacenado para la carga o inserción de los clientes con toda la

información correspondiente. Este procedimiento debe invocar una **Función** que

calcule la edad de los clientes.

SQL> Create or Replace Function Edad\_Calc (Nac\_Fecha Clientes.Fecha\_Birth%Type)

2

3 Return Number AS

4

5 Begin

6 Return Floor (Months\_Between (Sysdate, Nac\_Fecha)/12);

7 End Edad\_Calc;

8 /

Function created.

SQL> Create or Replace Procedure AddClientes (

2 p\_IDCliente in Clientes.ID\_Cliente%Type,

3 p\_Cedula IN Clientes.Cedula%Type,

4 p\_Nombre IN Clientes.Nombre%Type,

5 p\_Apellido in Clientes.Apellido%Type,

6 p\_Sexo IN Clientes.Sexo%Type,

7 p\_Birth IN Clientes.Fecha\_Birth%Type,

8 p\_CodProfesion IN Clientes.Cod\_Profesion%Type,

9 Cod\_Sucursal IN Clientes.Cod\_Sucursal%Type,

10 p\_Error OUT Varchar2

11 ) AS

12

13 p\_Edad Clientes.Edad%Type;

14

15 Begin

16

17 p\_error := 'Cliente Creado';

18 p\_Edad := Edad\_Calc(p\_Birth);

19

20 Insert INTO Clientes (ID\_Cliente, Cedula, Nombre, Apellido, Sexo, Fecha\_Birth,

21 Cod\_Profesion, Cod\_Sucursal)

22 Values(p\_IDCliente, p\_Cedula, p\_Nombre, p\_Apellido, p\_Sexo, p\_Birth,

23 p\_CodProfesion, Cod\_Sucursal);

24 Exception

25

26 ---When Dup\_Val\_On\_Index Then

27 ---p\_Error := 'Cliente Existente'

28

29 When Others Then

30 p\_Error := 'Cliente Creado';

31

32 Commit;

33 End AddClientes;

34 /

Procedure created.

 **Procedimiento** almacenado para la carga o inserción de los préstamos aprobados

con toda la información correspondiente e igualmente este proceso deberá

actualizar la información de préstamo en la tabla de sucursales.

SQL> create Or Replace Procedure ADDPrestamo (

p\_NPrestamo IN Prestamo.N\_Prestamo%Type,

p\_IDCliente IN Prestamo.ID\_Cliente%Type,

p\_CodPrestamo IN Prestamo.Cod\_Prestamo%Type,

p\_FechaAproved IN Prestamo.Fecha\_Aprobada%Type,

p\_MontoAproved IN Prestamo.Monto\_Aprobado%Type,

p\_TasaInteres IN Prestamo.Tasa\_Interes%Type,

p\_LetraMen IN Prestamo.Letra\_Mensual%Type,

p\_MontoPagado IN Prestamo.Monto\_Pagado%Type,

p\_FechaPagada IN Prestamo.Fecha\_Pagada%Type,

p\_SaldoActual IN Prestamo.SaldoActual%Type,

p\_FechaModf IN Prestamo.FechaModificacion%Type,

p\_InteresPagado IN Prestamo.InteresPagado%Type,

p\_Usuario IN Prestamo.Usuario%Type,

p\_CodSucursal IN Prestamo.Cod\_Sucursal%Type,

p\_Error Out Varchar2

) AS

Begin

p\_Error := 'Prestamo Creado';

Insert INTO Prestamo (N\_Prestamo, ID\_Cliente, Cod\_Prestamo, Fecha\_Aprobado, Monto\_Aprobado, Tasa\_Interes, Letra\_Mensual, Monto\_Pagado, Fecha\_Pagada, SaldoActual, InteresPagado, FechaModificacion, Usuario, Cod\_Sucursal)

Values (p\_NPrestamo, p\_IDCliente, p\_CodPrestamo, p\_FechaAproved, p\_MontoAproved, p\_TasaInteres, p\_LetraMen, p\_MontoPagado, p\_FechaPagada, p\_SaldoActual, p\_InsteresPagado, p\_FechModf, p\_Usuario, p\_CodSucursal);

Update SucursalTipoPres Set Monto\_Prestamo := Monto\_prestamo + Monto\_Aprobado

Where Cod\_Prestamo := p\_CodPrestamo AND Cod\_Sucursal := p\_CodSucursal;

Exception

When No\_Data\_Found Then

p\_Error := 'Prestamo no existente';

Commit;

End AddPrestamo;

/

**Procedimiento** almacenado para la carga o inserción de los pagos recibidos de los

clientes que se almacena en la tabla transacpagos. Por lo menos uno para cada

tipo de préstamos.

CREATE OR REPLACE PROCEDURE AddPago(

2 p\_IdTransac Tansacpago.ID\_Transaccion%TYPE,

3 p\_codSuc Tansacpago.Cod\_Sucursal%TYPE,

4 p\_idClien Tansacpago.ID\_Cliente%TYPE,

5 p\_ftransa Tansacpago.Fecha\_Transaccion%TYPE,

6 p\_finser Tansacpago.FechaInsercion%TYPE,

7 p\_montpa Tansacpago.MontoPago%TYPE,

8 p\_usuario Tansacpago.Usuario%TYPE,

9 p\_TPresta Tansacpago.TipoPrestamo%TYPE,

10 p\_error out varchar2)AS

11

12 BEGIN

13 p\_error := 'Proceso completado';

14 Insert INTO Tansacpagos (ID\_Transaccion,Cod\_Sucursal,ID\_Cliente,Fecha\_Transaccion,FechaInsercion,MontoPago,Usuario,TipoPrestamo)

15 VALUES(p\_IdTransac,p\_codSuc,p\_idClien,p\_ftransa,p\_finser,p\_montpa,p\_usuario,p\_TPresta);

16

17 EXCEPTION

18 WHEN DUP\_VAL\_ON\_INDEX THEN

19 p\_error:='El pago ya existe';

20 WHEN OTHERS THEN

21 p\_error:='No se logro crear el registro';

22 COMMIT;

23

24 END AddPago;

25 /

Warning: Procedure created with compilation errors.

SQL> Show Errors

Errors for PROCEDURE ADDPAGO:

LINE/COL ERROR

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

14/1 PL/SQL: SQL Statement ignored

14/13 PL/SQL: ORA-00942: table or view does not exist