Universidad Tecnológica de Panamá

Facultad de Ingeniería de Sistemas Computacionales

Licenciatura en Ingeniería de Sistemas de Información

Departamento de Sistemas de Información, Control y Evaluación de Recursos Informáticos

Sistemas de Bases de Datos II

Equipo #1

Laboratorio #5: “Implementación de Modelo Lógico Relacional Transaccional”

Facilitador:

Ing. Henry Lezcano

Integrantes:

Aguilar, Milagros 3-740-771

Atencio, Anel 8-950-868

Márquez, Paola 8-949-1108

Rojas, Reynaldo 8-950-792

Grupo:

1IF131

II Semestre, 2020

1. **ENUNCIADO:**

La Compañía Financiera Márquez-Cedeño, S.A., necesita levantar un modelo físico  
de datos a partir de un modelo lógico relacional para una sección de su proceso de  
negocio (otorgamiento de préstamos), en lo corresponde a la entrada de la  
información de los cliente y los préstamos e información relacionada a esta sección del proceso a dicha base de datos. Cuenta con unas relaciones o tablas de bases de datos que han sido identificadas y no están normalizadas que pueden ser utilizada por los Ingenieros de Sistemas de Información para futuras implementación.

La relación Cliente está compuesta por identificación del cliente, cedula, nombre, apellido, sexo, emial (personal, laboral, académico), teléfono (celular, residencia, celular del familiar más cercano, celular del conyugue), profesión (cualquier profesión del cliente), fecha de nacimiento.

La relación Préstamo está compuesta por identificación del cliente, tipo de  
préstamo (personal, auto, hipoteca, garantizado con ahorros), número de  
préstamos, fecha de aprobado, monto aprobado, tasa de interés, letra mensual,  
monto pagado, fecha de pago. Los clientes pueden tener varios tipos de préstamos  
en la institución financiera.

* Muestre el modelo Lógico relacional normalizado
* Muestre el Script (DDL) que corresponde al levantamiento del modelo físico.
* Levante el Modelo Físico a partir del modelo Lógico relacional normalizado  
  usando su script creado cumpliendo con
  + Las restricciones inherente al modelo
  + Las restricciones semánticas o del usuario

Se pide una vez el esquema de Base de Datos este levantado en el sistema de  
gestión, la información de la Base de Datos también quede levantada usando un  
script (DML) para su evaluación.

1. **Modelo Lógico Relacional Normalizado**











1. **Código de la creación de tablas – Script DDL (Modelo Físico)**

create table PROFESION (

ID\_Pro number not null,

profesion varchar2 (30) not null,

constraint ID\_Pro\_pk primary key (ID\_Pro)

);

create table TIPO\_PRESTAMO (

ID\_tipo number not null,

tipo varchar2 (30) not null,

tasa\_interes number not null,

constraint ID\_tipo\_pk primary key (ID\_tipo)

);

create table EMAIL (

ID\_email number not null,

Tipo\_email varchar2 (20) not null,

constraint ID\_email\_pk primary key (ID\_email)

);

create table TELEFONO (

ID\_telefono number not null,

Tipo\_telefono varchar2 (20) not null,

constraint ID\_telefono\_pk primary key (ID\_telefono)

);

create table CLIENTE (

ID\_Cliente number not null,

ID\_Pro number not null,

cedula varchar2 (20) not null,

nombre varchar2 (30) not null,

apellido varchar2 (30) not null,

sexo char not null,

fecha\_nac date not null,

constraint ID\_Cliente\_pk primary key (ID\_Cliente),

constraint ID\_prof\_fk foreign key (ID\_Pro) references PROFESION (ID\_Pro)

);

create table EMAIL\_CLIENTE (

ID\_email number not null,

ID\_Cliente number not null,

email varchar2 (40) not null,

constraint ID\_email\_fk foreign key (ID\_email) references EMAIL (ID\_email),

constraint ID\_Cl\_email\_fk foreign key (ID\_Cliente) references CLIENTE (ID\_Cliente),

constraint ID\_email\_cl\_pk primary key (ID\_email, ID\_Cliente)

);

create table TELEFONO\_CLIENTE (

ID\_telefono number not null,

ID\_Cliente number not null,

telefono varchar2 (20) not null,

constraint ID\_tel\_fk foreign key (ID\_telefono) references TELEFONO (ID\_telefono),

constraint ID\_cl\_tel\_fk foreign key (ID\_Cliente) references CLIENTE (ID\_Cliente),

constraint ID\_telef\_cl\_pk primary key (ID\_telefono, ID\_Cliente)

);

create table PRESTAMO (

ID\_tipo number not null,

ID\_Cliente number not null,

Num\_prestamo number not null,

fecha\_aprob date not null,

monto\_aprob number not null,

letra\_mensual number not null,

monto\_pago number not null,

fecha\_pago date not null,

constraint ID\_tipo\_fk foreign key (ID\_tipo) references TIPO\_PRESTAMO (ID\_tipo),

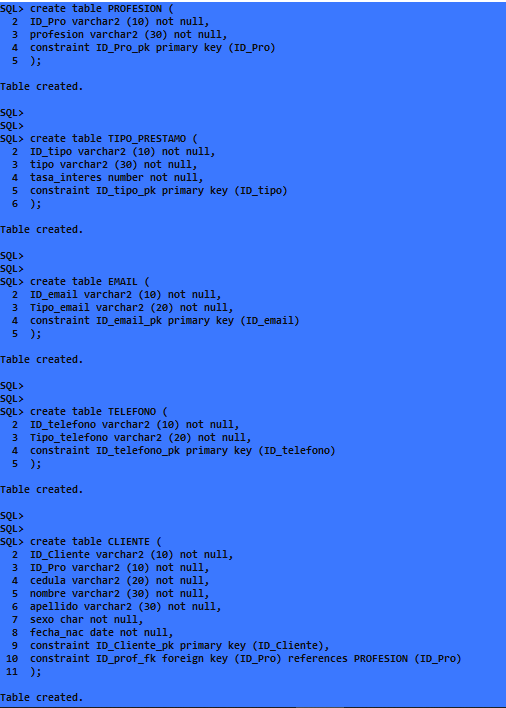
constraint ID\_cliente\_prest\_fk foreign key (ID\_Cliente) references CLIENTE (ID\_Cliente),

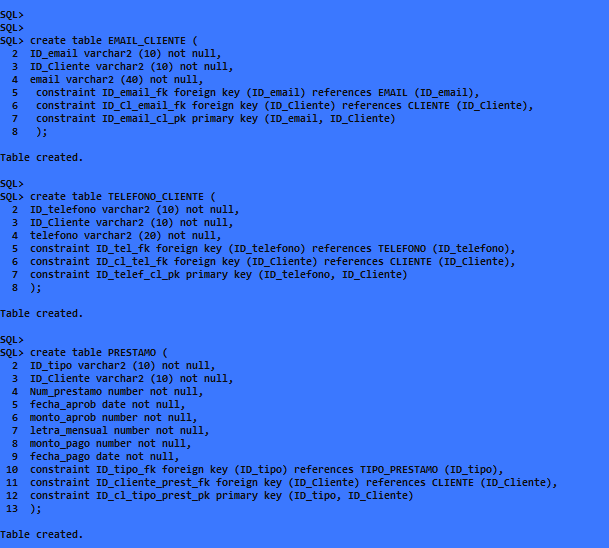
constraint ID\_cl\_tipo\_prest\_pk primary key (ID\_tipo, ID\_Cliente)

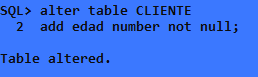
);

create unique index numero\_prestamo on PRESTAMO (Num\_prestamo);

1. **Levantamiento del modelo físico**







alter table CLIENTE

add edad number not null;

create table SUCURSAL (

Cod\_sucursal number not null,

Nombre\_sucursal varchar2 (30) not null,

Constraint Cod\_suc\_pk primary key (Cod\_sucursal)

);

create table SUCURSAL\_PRESTAMO(

Cod\_sucursal number not null,

ID\_tipo number not null,

Constraint ID\_tipo\_suc\_fk foreign key (ID\_tipo) references TIPO\_PRESTAMO (ID\_tipo),

Constraint Cod\_sucursal\_p\_fk foreign key (Cod\_sucursal) references SUCURSAL (Cod\_sucursal),

Constraint Cod\_id\_suc\_pk primary key (Cod\_sucursal, ID\_tipo),

Monto\_prestamo number not null

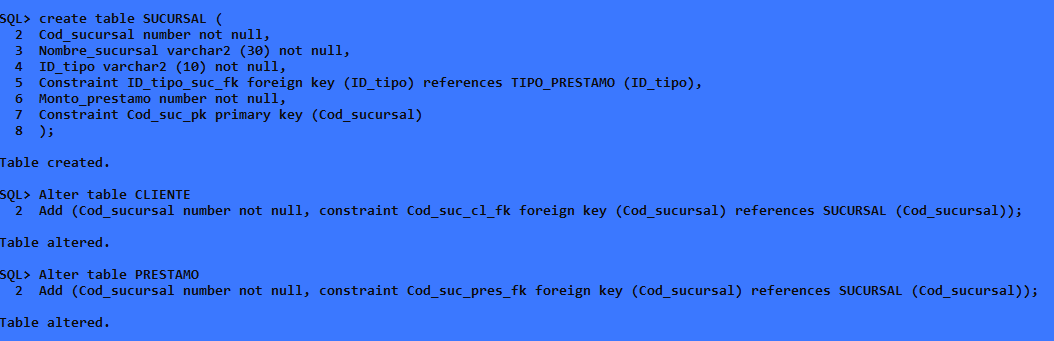
);

Alter table CLIENTE

Add (Cod\_sucursal number not null, constraint Cod\_suc\_cl\_fk foreign key (Cod\_sucursal) references SUCURSAL (Cod\_sucursal));

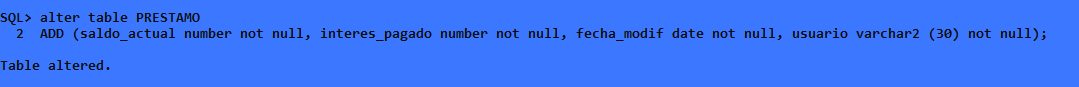
Alter table PRESTAMO

Add (Cod\_sucursal number not null, constraint Cod\_suc\_pres\_fk foreign key (Cod\_sucursal) references SUCURSAL (Cod\_sucursal));



alter table PRESTAMO

ADD (saldo\_actual number not null, interes\_pagado number not null, fecha\_modif date not null, usuario varchar2 (30) not null);



create table transact\_pagos (

ID\_transaccion number not null,

constraint ID\_transaccion\_pk primary key (ID\_transaccion),

Cod\_sucursal number not null,

ID\_Cliente number not null,

ID\_tipo number not null,

Fechatransaccion date not null,

Monto\_pago number not null,

Fecha\_insercion date not null,

Usuario varchar2 (30) not null

);

CREATE SEQUENCE email\_sequence

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

CREATE OR REPLACE PROCEDURE add\_email (

p\_id\_email IN email.id\_email%TYPE,

p\_tipo\_email IN email.tipo\_email%TYPE

) AS

BEGIN

INSERT INTO email (id\_email, tipo\_email)

VALUES (email\_sequence.nextval, p\_tipo\_email);

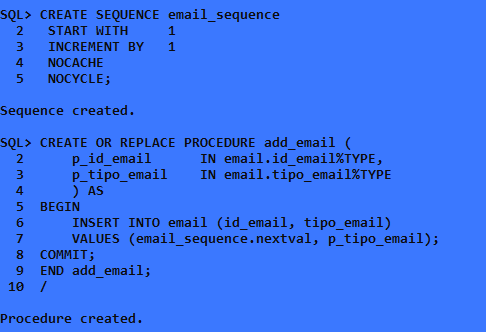
EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE (‘No se encontró la tabla’);

COMMIT;

END add\_email;



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

CREATE SEQUENCE telefono\_sequence

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

CREATE OR REPLACE PROCEDURE add\_telefono (

p\_id\_telefono IN telefono.id\_telefono%TYPE,

p\_tipo\_telefono IN telefono.tipo\_telefono%TYPE

) AS

BEGIN

INSERT INTO telefono (id\_telefono, tipo\_telefono)

VALUES (telefono\_sequence.nextval, p\_tipo\_telefono);

COMMIT;

END add\_telefono;

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

CREATE SEQUENCE profesion\_sequence

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

CREATE OR REPLACE PROCEDURE add\_profesion (

p\_id\_pro IN profesion.id\_pro%TYPE,

p\_profesion IN profesion.profesion%TYPE

) AS

BEGIN

INSERT INTO profesion (id\_pro, profesion)

VALUES (profesion\_sequence.nextval, p\_profesion);

COMMIT;

END add\_profesion;

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

CREATE SEQUENCE tipo\_prestamo\_sequence

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

CREATE OR REPLACE PROCEDURE add\_tipo\_prestamo (

p\_id\_tipo IN tipo\_prestamo.id\_tipo%TYPE,

p\_tipo IN tipo\_prestamo.tipo%TYPE,

p\_tasa\_interes IN tipo\_prestamo.tasa\_interes%TYPE

) AS

BEGIN

INSERT INTO tipo\_prestamo (id\_tipo, tipo, tasa\_interes)

VALUES (tipo\_prestamo\_sequence.nextval, p\_tipo, p\_tasa\_interes);

COMMIT;

END add\_tipo\_prestamo;

**-----------------------------------------------------------**

**Función para calcular edad**

CREATE OR REPLACE FUNCTION calc\_edad (fecha\_nac CLIENTE.fecha\_nac%TYPE)

RETURN number AS

edad NUMBER;

BEGIN

SELECT (months\_between (sysdate, fecha\_nac)/12 ) INTO edad from CLIENTE;

RETURN edad;

END calc\_edad;