# Temática del proyecto:

# En el presente proyecto se trabajara la base de datos de un banco de la Republica Argentina.

# Tablas de un banco:

ID\_Tarjeta (pk)

Tipo Tarjeta

Marca Tarjeta

ID\_Cuenta (fk)

Limite máx.

Limite Disponible

Fecha de Vencimiento

Tarjetas

ID\_Transaccion (pk) Descripcion

ID\_Cuenta (fk)

fecha

Estado

Transacciones

ID\_Cuenta(pk) Tipo Cuenta Limite Cuenta Cantidad de tarjeta de credito

Cuenta

ID\_Cliente (pk) Nombre Apellido

ID\_Cuenta(fk) ID\_Sucursal (fk)

Cliente

ID\_Empleado (pk) ID\_Sucursal (fk) Nombre

Apellido

Sueldo

Antigüedad

Empleados

ID\_Cajero (pk) ID\_Sucursal (fk) Direccion

Saldo Actual

Cajeros

ID\_Sucursal(pk) Nombre Direccion

Sucursal

# Tabla Transacciones:

*Descripción:* En la tabla transacciones se almacenan todas las transacciones registradas.

*Campos:*

|  |  |
| --- | --- |
| *Nombre Campo* | *Tipo de dato* |
| *ID\_Transaccion* | *Entero (int)* |
| *Descripcion* | *Caracter (string)* |
| *ID\_Cuenta* | *Entero (int)* |
| *Fecha* | *Fecha (date)* |
| *Estado* | *Boolean (bool)* |

# Tabla Sucursal:

*Descripción:* En la tabla transacciones se almacenan todas las sucursales operativas.

*Campos:*

|  |  |
| --- | --- |
| *Nombre Campo* | *Tipo de dato* |
| *ID\_Sucursal* | *Entero (int)* |
| *Nombre* | *Caracter (string)* |
| *Direccion* | *Carácter (string)* |

Tabla Cliente:

*Descripción:* En la tabla transacciones se almacenan toda la información de los clientes del banco.

*Campos:*

|  |  |
| --- | --- |
| *Nombre Campo* | *Tipo de dato* |
| *ID\_Cliente* | *Entero (int)* |
| *Nombre* | *Caracter (string)* |
| *Apellido* | *Caracter (string)* |
| *ID\_Cuenta* | *Entero (int)* |
| *ID\_Sucursal* | *Entero (int)* |

# Tabla Empleados:

*Descripción:* En la tabla transacciones se almacenan toda la información de los empleados del banco.

*Campos:*

|  |  |
| --- | --- |
| *Nombre Campo* | *Tipo de dato* |
| *ID\_Empleado* | *Entero (int)* |
| *Nombre* | *Caracter (string)* |
| *Apellido* | *Caracter (string)* |
| *ID\_Sucursal* | *Entero (int)* |
| *Sueldo* | *Decimal (float)* |
| *Antigüedad* | *Entero (int)* |

# Tabla Cuenta:

*Descripción:* En la tabla transacciones se almacenan toda la información de las cuentas de los clientes del banco.

*Campos:*

|  |  |
| --- | --- |
| *Nombre Campo* | *Tipo de dato* |
| *ID\_Cuenta* | *Entero (int)* |
| *Tipo Cuenta* | *Caracter (string)* |
| *Limete Cuenta* | *Decimal (float)* |
| *Cantida de Tarjetas de Credito* | *Entero (int)* |

# Tabla Tarjetas:

*Descripción:* En la tabla transacciones se almacenan todas las tarjetas de los clientes del banco.

*Campos:*

|  |  |
| --- | --- |
| *Nombre Campo* | *Tipo de dato* |
| *ID\_Tarjeta* | *Entero (int)* |
| *Tipo Tarjeta* | *Caracter (string)* |
| *Marca Tarjeta* | *Carácter (string)* |
| *ID\_Cuenta* | *Entero (int)* |
| *Limite Max* | *Decimal (float)* |
| *Limite Disponible* | *Decimal (float)* |
| *Fecha de Vencimiento* | *Fecha (date)* |

# Tabla Cajeros Automáticos (estaciones de servicio al cliente):

*Descripción:* En la tabla transacciones se almacenan toda la información de los cajeros automáticos que posee el banco distribuidos por el país.

*Campos:*

|  |  |
| --- | --- |
| *Nombre Campo* | *Tipo de dato* |
| *ID\_Cajero* | *Entero (int)* |
| *ID\_Sucursal* | *Entero (int)* |
| *Direccion* | *Carácter (string)* |
| *Saldo Actual* | *Decimal (float)* |

# Vistas Creadas

# En el presente trabajo se crearon las siguientes vistas:

# clientes\_cuenta 🡪 La misma cumple el objetivo de facilitar el acceso de la información tanto del cliente como de su cuenta con una simple consulta.

# transacciones\_aprobadas 🡪 La misma cumple la función se visualizarnos solamente aquellas transacciones que han sido aprobadas.

# clientes\_tarjetas 🡪 La misma cumple el objetivo de facilitar el acceso de la información tanto del cliente como de sus tarjetas con una simple consulta.

# empleados\_sucursal 🡪 La misma cumple el objetivo de facilitar el acceso de la información tanto de los empleados como de la sucursal en la que operan con una simple consulta.

# empleados\_con\_antiguedad 🡪 La misma cumple la función de visualizarnos solo aquellos empleados que poseen antigüedad en el banco.

# Funciones Creadas

# En el presente trabajo se crearon las siguientes funciones:

# calculateSalararioConAntiguedad 🡪 La misma calcula el salario de los empleados incluyendo un adicional por antigüedad.

# nuevo\_limite\_max 🡪 La misma calcula el Nuevo limite máximo permitido por tarjeta

# Procedimientos Creados

# En el presente trabajo se crearon los siguientes procedimientos:

# - sort\_tabla\_cuenta 🡪 Procedimiento que ordena la tabla cuenta en función del campo que se desee ordenar y la definición si es asc o des. El mismo recibe como parámetro un campo de la tabla cuenta y el tipo de orden que se requiere.

# - insert\_data\_sucursal 🡪 El mismo incorpora datos a la tabla sucursal. Se requieren como parámetro los valores a ser insertados.

# MySQL:

# CREATE schema PROYECTO;

# use PROYECTO;

# CREATE TABLE cuenta (

# id\_cuenta int not null,

# tipo\_cuenta varchar(500) not null,

# limite\_cuenta float,

# cant\_tarjetas int,

# PRIMARY KEY (id\_cuenta)

# );

# CREATE TABLE tarjetas (

# id\_tarjeta int not null,

# tipo\_tarjeta varchar(500) not null,

# marca\_tarjeta varchar(500) not null,

# id\_cuenta int not null,

# limite\_max float,

# limite\_disponible float,

# fecha\_venc DATE,

# PRIMARY KEY (id\_tarjeta),

# foreign key (id\_cuenta) references cuenta (id\_cuenta)

# );

# CREATE TABLE transacciones (

# id\_transaccion int not null,

# descripcion varchar(500) not null,

# id\_cuenta int not null,

# fecha DATE,

# estado bool,

# PRIMARY KEY (id\_transaccion),

# foreign key (id\_cuenta) references cuenta (id\_cuenta)

# );

# CREATE TABLE sucursal (

# id\_sucursal int not null,

# nombre varchar(500) not null,

# direccion varchar(500) not null,

# PRIMARY KEY (id\_sucursal)

# );

# CREATE TABLE empleados (

# id\_empleado int not null,

# id\_sucursal int not null,

# nombre varchar(500) not null,

# apellido varchar(500) not null,

# sueldo float not null,

# antiguedad int,

# PRIMARY KEY (id\_empleado),

# foreign key (id\_sucursal) references sucursal (id\_sucursal)

# );

# CREATE TABLE cajeros (

# id\_cajero int not null,

# id\_sucursal int not null,

# direccion varchar(500) not null,

# saldo\_actual float,

# PRIMARY KEY (id\_cajero),

# foreign key (id\_sucursal) references sucursal (id\_sucursal)

# );

# CREATE TABLE clientes (

# id\_cliente int not null,

# id\_cuenta int not null,

# id\_sucursal int not null,

# nombre varchar(500) not null,

# apellido varchar(500) not null,

# PRIMARY KEY (id\_cliente),

# foreign key (id\_sucursal) references sucursal (id\_sucursal),

# foreign key (id\_cuenta) references cuenta (id\_cuenta)

# );

# insert into cuenta (id\_cuenta,tipo\_cuenta,limite\_cuenta,cant\_tarjetas) values (1, 'gold',20000,3);

# insert into cuenta (id\_cuenta,tipo\_cuenta,limite\_cuenta,cant\_tarjetas) values (2, 'clasic',10000,1);

# insert into tarjetas (id\_tarjeta,tipo\_tarjeta,marca\_tarjeta,id\_cuenta,limite\_max,limite\_disponible) values (1,'credito','visa',1, 15000,10000);

# insert into transacciones (id\_transaccion,descripcion,id\_cuenta,fecha,estado) values (1,'retiro efectivo cajero',1,01/01/2022,1);

# insert into sucursal (id\_sucursal,nombre,direccion) values (1,'casa central','paraguay 777');

# insert into empleados (id\_empleado,id\_sucursal,nombre,apellido,sueldo,antiguedad) values(1,1,'Rocio','Vega',1000,2);

# insert into cajeros (id\_cajero,id\_sucursal,direccion,saldo\_actual) values(1,1,'paraguay 777',3000000);

# insert into clientes (id\_cliente,id\_cuenta,id\_sucursal,nombre,apellido) values(1,1,1,'Belen','Vega');

# INSERT INTO cuenta (id\_cuenta,tipo\_cuenta,limite\_cuenta,cant\_tarjetas)

# VALUES

# (3,"clasic",53,2),

# (4,"black",465,1),

# (5,"black",428,3),

# (6,"clasic",18,0),

# (7,"clasic",839,3),

# (8,"black",493,4),

# (9,"clasic",996,4),

# (10,"clasic",922,1),

# (12,"clasic",77,1),

# (11,"gold",412,1);

# INSERT INTO cuenta (id\_cuenta,tipo\_cuenta,limite\_cuenta,cant\_tarjetas)

# VALUES

# (13,"gold",734,1),

# (14,"clasic",418,2),

# (15,"gold",114,4),

# (16,"clasic",956,5),

# (17,"clasic",954,2),

# (18,"gold",764,3),

# (19,"clasic",161,3),

# (20,"gold",801,2),

# (21,"clasic",459,2),

# (22,"black",745,3);

# INSERT INTO cuenta (id\_cuenta,tipo\_cuenta,limite\_cuenta,cant\_tarjetas)

# VALUES

# (23,"clasic",5,3),

# (24,"clasic",126,5),

# (25,"gold",218,1),

# (26,"clasic",461,4),

# (27,"black",496,4),

# (28,"black",278,4),

# (29,"black",274,1),

# (30,"black",735,1),

# (31,"clasic",269,5),

# (32,"black",234,4);

# INSERT INTO tarjetas (id\_tarjeta,tipo\_tarjeta,marca\_tarjeta,id\_cuenta,limite\_max,limite\_disponible)

# VALUES

# (2,"credito","visa",1,6059,4410),

# (3,"credito","mastercard",2,9200,2485),

# (4,"credito","visa",3,5034,103),

# (5,"credito","mastercard",4,9939,2071),

# (6,"credito","mastercard",5,42090,7754),

# (7,"credito","visa",6,9957,1586),

# (8,"credito","mastercard",7,60500,9559),

# (9,"credito","mastercard",8,40200,6789),

# (10,"credito","visa",9,9969,4101),

# (11,"credito","mastercard",10,55190,9430);

# INSERT INTO tarjetas (id\_tarjeta,tipo\_tarjeta,marca\_tarjeta,id\_cuenta,limite\_max,limite\_disponible)

# VALUES

# (12,"credito","mastercard",11,10570,3001),

# (13,"credito","visa",12,25609,8124),

# (14,"credito","mastercard",13,9227,2758),

# (15,"credito","visa",14,9819,1301),

# (16,"credito","visa",15,2780,465),

# (17,"credito","mastercard",16,6619,694),

# (18,"credito","mastercard",17,6463,602),

# (19,"credito","visa",18,1725,947),

# (20,"credito","mastercard",19,4815,521),

# (21,"credito","visa",20,8753,6023);

# INSERT INTO transacciones (id\_transaccion,descripcion,id\_cuenta,fecha,estado)

# VALUES

# (2,"solicitud de prestado",1,01/07/2021,"1"),

# (3,"solicitud de prestado",2,11/08/2021,"1"),

# (4,"solicitud de prestado",3,10/08/2021,"1");

# insert into sucursal (id\_sucursal,nombre,direccion)

# values

# (2,'centro','entre rios 1200'),

# (3,'Fisherton','eva peron 600');

# INSERT INTO empleados (id\_empleado,id\_sucursal,nombre,apellido,sueldo,antiguedad)

# VALUES

# (2,2,"Rowan","Lacy",9459,1),

# (3,2,"Akeem","Jenna",8240,7),

# (4,3,"Hashim","Serina",6304,10),

# (5,2,"Yen","Macaulay",9251,23),

# (6,1,"Rashad","Herman",5508,0);

# INSERT INTO clientes (id\_cliente,id\_cuenta,id\_sucursal,nombre,apellido)

# VALUES

# (2,1,1,"Quinlan","Isaac"),

# (3,2,3,"Karen","Yuli"),

# (4,3,3,"Lillith","Tarik"),

# (5,4,2,"Acton","Kyla"),

# (6,5,2,"Lyle","Walker");

# create view clientes\_cuenta as

# (select cl.nombre, cl.apellido, cl.id\_cuenta, c.cant\_tarjetas, c.limite\_cuenta, c.tipo\_cuenta

# from clientes as cl

# inner join cuenta as c

# on cl.id\_cuenta=c.id\_cuenta);

# create view transacciones\_aprobadas as

# (select \*

# from transacciones

# where estado=1);

# create view clientes\_tarjetas as

# (select c.nombre, c.apellido, c.id\_cuenta, t.marca\_tarjeta, t.tipo\_tarjeta, t.limite\_disponible, t.limite\_max

# from clientes as c

# inner join tarjetas as t

# on c.id\_cuenta=t.id\_cuenta);

# create view empleados\_sucursal as

# (select e.nombre, e.apellido, e.antiguedad, s.direccion

# from empleados as e

# inner join sucursal as s

# on e.id\_sucursal=s.id\_sucursal);

# create view empleados\_con\_antiguedad as

# (select \*

# from empleados as e

# where e.antiguedad>0);

# CREATE FUNCTION `calculateSalararioConAntiguedad`(antiguedad INT, salario DECIMAL) RETURNS DECIMAL DETERMINISTIC

# BEGIN

# DECLARE fullSalary float;

# SET fullSalary = salario+salario\*antiguedad\*0.1;

# RETURN fullsalary;

# END;

# CREATE FUNCTION `nuevo\_limite\_max`(limite\_max float) RETURNS float DETERMINISTIC

# BEGIN

# DECLARE nuevo\_limite\_max float;

# SET nuevo\_limite\_max = limite\_max\*1.5;

# RETURN nuevo\_limite\_max;

# END;

# -- PROCEDIMIENTO QUE ORDENA TABLA CUENTA EN FUNCION DEL CAMPO PASADO COMO PARAMETRO Y EL TIPO DE ORDENAMIENTO PASADO COMO PARAMETRO

# DELIMITER &&

# CREATE PROCEDURE sort\_tabla\_cuenta (IN campo\_order varchar (255), IN tipo\_orden varchar(255) )

# BEGIN

# SET @query = CONCAT ('SELECT \*

# FROM cuenta ORDER BY (',campo\_order,')',tipo\_orden);

# PREPARE stmt FROM @query;

# EXECUTE stmt;

# DEALLOCATE PREPARE stmt;

# END &&

# DELIMITER ;

# -- PROCEDIMIENTO QUE INSERTA DATA EN SUCURSAL

# DELIMITER &&

# CREATE PROCEDURE insert\_data\_sucursal (IN id\_sucursal\_new int, IN nombre\_new varchar(255), in direccion\_new varchar(255) )

# BEGIN

# insert into sucursal(id\_sucursal, nombre, direccion) values (id\_sucursal\_new, nombre\_new,direccion\_new);

# END &&

# DELIMITER ;

# drop table if exists Log\_auditoria;

# Create table if not exists Log\_auditoria

# (id\_log int auto\_increment,

# nombre\_accion varchar (20),

# nombre\_tabla varchar (20),

# usuario varchar (20),

# fecha date,

# primary key (id\_log));

# DELIMITER &&

# Create trigger TRG\_LOG\_EMPLEADOS after insert on empleados

# for each row

# begin

# insert into Log\_auditoria (nombre\_accion, nombre\_tabla, usuario, fecha)

# values ('INSERT','empleados',current\_user(),now());

# end &&

# DELIMITER ;

# DELIMITER &&

# Create trigger TRG\_LOG\_EMPLEADOS\_update after update on empleados

# for each row

# begin

# insert into Log\_auditoria (nombre\_accion, nombre\_tabla, usuario, fecha)

# values ('UPDATE','empleados',current\_user(),now());

# end &&

# DELIMITER ;

# drop table if exists Movimientos\_auditoria;

# Create table if not exists Movimientos\_auditoria

# (id\_log int auto\_increment,

# nombre\_accion varchar (20),

# nombre\_tabla varchar (20),

# usuario varchar (20),

# fecha date,

# campo\_modificado varchar (20),

# primary key (id\_log));

# DELIMITER &&

# Create trigger TRG\_LOG\_clientes\_update after update on clientes

# for each row

# begin

# insert into Movimientos\_auditoria (nombre\_accion, nombre\_tabla, usuario, fecha, hora, campo\_modificado)

# values ('UPDATE','clientes',current\_user(),now(),current\_time(),old.nombre);

# end &&

# DELIMITER ;

# DROP TABLE IF EXISTS cuentaArchivos;

# CREATE TABLE cuentaArchivos (

# id\_cuenta INT PRIMARY KEY AUTO\_INCREMENT,

# numero\_cuenta INT,

# deletedAt TIMESTAMP DEFAULT NOW()

# );

# DELIMITER $$

# CREATE TRIGGER before\_cuenta\_delete

# BEFORE DELETE

# ON cuenta FOR EACH ROW

# BEGIN

# INSERT INTO cuentaArchivos(numero\_cuenta)

# VALUES(OLD.id\_cuenta);

# END$$

# DELIMITER ;