

DESARROLLE IMPLEMENTACION DE PRORAMACION ALMACENDA

Para el proceso de negocio puesto en producción en el Laboratorio No.6, realice los siguientes cambios solicitados por el equipo de analistas. Estos cambios se harán usando el Modelo Físico Normalizado e implementado previamente (*Puede usar la tabla de auditoria del Laboratorio 7 si el Laboratorio 6 no la tuviera*)

- Un triggers para el registro en la tabla de auditoria que guarde la transacción que corresponde a la aprobación de los créditos o préstamos.
- Un triggers para cargar los montos de los préstamos aprobados en la tabla que acumula los saldo de los préstamos por sucursal y tipo de préstamos.
- Un triggers que registre en la tabla de auditoria la aplicaciones de las transacciones de pagos al prestamos , reversiones o eliminaciones
- Un triggers que registre las afectaciones de las transacciones de pagos a los préstamos, reversiones o eliminaciones en la tabla que acumula los saldo de los préstamos por sucursal y tipo de préstamos.
- Una vista de los préstamos por sucursal y tipo de esta forma:

Préstamo por Sucursal y por tipo

sucursal	Tipo Préstamo	monto
El dorado	corriente	5000.00
pedregal	especial	2000.00

--CREACIÓN DE LA TABLA CLIENTE PRESTAMO

Create table Clientes_Prestamo (

Numero_prestamo varchar2(10) not null ,

Fecha_aprob date not null,

Monto_aprob number,

Letra_mensual number not null,

Cod_Prestamo,

Constraint codigo_prestamo foreign key (Cod_Prestamo) references
Prestamo_Tipo(Cod_Prestamo),

ID_cliente,

Constraint identif_cliente_prestamo foreign key (ID_Cliente) references Banco_Clientes
(ID_Cliente),

Constraint Presta_Cliente primary key (Cod_Prestamo , ID_Cliente)

);

--CREACIÓN DE LA TABLA SUCURSAL

```
Create table Sucursal (  
Cod_Sucursal varchar2(5) primary key not null,  
Nom_Sucursal varchar2(25),  
Cod_Prestamo varchar2(15),  
Constraint Fk_cod_prestamo foreign key (Cod_Prestamo) references  
Prestamo_Tipo (Cod_Prestamo),  
Monto_Prestamos number(15,2),  
Constraint pk_tipo_suc_prest primary key (Cod_Sucursal, Cod_Prestamo)  
);
```

--CREACIÓN DE LA TABLA SUCURSAL TIPO PRESTAMO

```
Create table Suc_Tipo_Prestamo (  
Cod_Sucursal varchar2(5),  
Constraint Fk_cod_suc foreign key (Cod_Sucursal) references Sucursal  
(Cod_Sucursal),  
Cod_Prestamo varchar2(15),  
Constraint Fk_cod_prestamo foreign key (Cod_Prestamo) references  
Prestamo_Tipo (Cod_Prestamo),  
Monto_Prestamos number(15,2),  
Constraint pk_tipo_suc_prest primary key (Cod_Sucursal, Cod_Prestamo)  
);
```

--CREACIÓN DE LA TABLA AUDITORÍA

```
Create table Auditoria (  
Numero_prestamo varchar2(10) not null ,  
Fecha_aprob date not null,  
Monto_aprob number,  
Letra_mensual number not null,  
Fecha_audi date,  
Usuario varchar2(20),  
Cod_Prestamo,  
Constraint codigo_prestamo foreign key (Cod_Prestamo) references  
Prestamo_Tipo(Cod_Prestamo),
```

```

ID_cliente,
Constraint identif_cliente_prestamo foreign key (ID_Cliente) references
Banco_Clientes (ID_Cliente),
Constraint Presta_Cliente primary key (Cod_Prestamo , ID_Cliente)
);

```

---Creación de tabla Transac_pagos

```

Create table Transac_pagos (
ID_Transaccion number primary key not null,
Cod_Sucursal varchar2(5) not null,
Monto_pago number not null,
Fecha_transac date not null,
Fecha_inserc date not null,
Usuario number not null,
Constraint FK_CodSuc_TransP foreign key (Cod_Sucursal) references Sucursal
(Cod_Sucursal),
ID_Cliente number,
Cod_Prestamo varchar2(15),
Constraint FK_IDClient_TPrest_TransP foreign key (ID_Cliente, Cod_Prestamo)
references Clientes_Prestamo (ID_Cliente, Cod_Prestamo)
);

```

- Trigger que guarde la transacción de créditos o préstamos aprobados

```

CREATE OR REPLACE TRIGGER A_Prestamo_Aprob
BEFORE INSERT ON CLIENTES_PRESTAMO FOR EACH ROW

BEGIN
INSERT INTO AUDITORIA (REG_AUDITORIA, numero_prestamo, fecha_aprob,
monto_aprob, letra_mensual, cod_prestamo, Id_cliente)
VALUES (seq_reg_auditoria.NEXTVAL, :new.numero_prestamo, :new.fecha_aprob,
:new.monto_aprob, :new.letra_mensual, :new.cod_prestamo, :new.Id_cliente);
END A_Prestamo_Aprob;
/

```

- Trigger que guarde el monto de los préstamos en la tabla Suc_tipo_prestitos

```

CREATE OR REPLACE TRIGGER Sucursales_Prestamo
BEFORE INSERT ON CLIENTES_PRESTAMO FOR EACH ROW

BEGIN
INSERT INTO Suc_tipo_prestito (cod_sucursal, cod_prestamo, monto_aprob)
VALUES (:new.cod_sucursal, :new.cod_prestamo, :new.monto_aprob);
END Sucursales_Prestamo;
/

```

- TRIGGER PARA INSERTAR, ELIMINAR Y ACTUALIZAR TABLA Auditoria
CREATE OR REPLACE TRIGGER Pagos_Prestamos

BEFORE UPDATE OR DELETE OR INSERT ON Transac_pagos

FOR EACH ROW

DECLARE

v_TipoCambio CHAR(1);

BEGIN

IF UPDATING THEN

v_TipoCambio := 'A';

INSERT INTO AUDITORIA

VALUES (seq_reg_auditoria.NEXTVAL, :new.numero_prestamo,:new.fecha_aprob,
:new.monto_aprob, :new.letra_mensual,:new.cod_prestamo, :new.Id_cliente);

ELSIF INSERTING THEN

v_TipoCambio := 'B';

INSERT INTO AUDITORIA

VALUES (seq_reg_auditoria.NEXTVAL, :new.numero_prestamo,:new.fecha_aprob,
:new.monto_aprob, :new.letra_mensual,:new.cod_prestamo, :new.Id_cliente);

ELSIF DELETING THEN

v_TipoCambio := 'C';

INSERT INTO AUDITORIA

VALUES (seq_reg_auditoria.NEXTVAL, :new.numero_prestamo,:new.fecha_aprob,
:new.monto_aprob, :new.letra_mensual,:new.cod_prestamo, :new.Id_cliente);

END IF;

END Pagos_Prestamos;

/

- TRIGGER PARA INSERTAR, ELIMINAR Y ACTUALIZAR TABLA SUC_TIPO_PRESTAMO

```

CREATE OR REPLACE TRIGGER Sucursal_pagos
BEFORE UPDATE OR DELETE OR INSERT ON Transac_pagos
FOR EACH ROW
DECLARE
    v_TipoCambio CHAR(1);
BEGIN
    IF UPDATING THEN
        v_TipoCambio := 'A';
        INSERT INTO Suc_tipo_prestamo
        VALUES (:new.cod_sucursal,:new.cod_prestamo, :new.monto_aprob);
    ELSIF INSERTING THEN
        v_TipoCambio := 'B';
        INSERT INTO Suc_tipo_prestamo
        VALUES (:new.cod_sucursal,:new.cod_prestamo, :new.monto_aprob);
    ELSIF DELETING THEN
        v_TipoCambio := 'C';
        INSERT INTO Suc_tipo_prestamo
        VALUES (:new.cod_sucursal,:new.cod_prestamo, :new.monto_aprob);
    END IF;
    END Sucursal_pagos;
/

```

- Creación de vista

```

CREATE VIEW Sucursal_tipo As

```

```

Select Nom_sucursal s, tipo_prestamo p, monto_aprob s

```

```

From Sucursal s join Prestamo_tipo on s.cod_sucursal = p.cod_sucursal;

```