

*/\* El Banco Nacional de Mis Provincias, deseas que se implemente una Base de Datos en Oracle y un sistema de procesos almacenados (objetos como lo son Funciones, procedimiento, disparadores y secuencias). Adicionalmente dentro de los procedimientos y funciones los procesos sean implementados con cursores, ciclos de control y estructuras de control. \*/*

*/\* Las tablas para la base de datos son las siguientes:*

*Auditoria (aud\_registro number, aud\_tabla varchar2(30), aud\_operación varchar2(1), aud\_usuario varchar2(30), aud\_fecha date)*

*\* (La llave primaria es el (aud\_registro) donde registro es un numero secuencia que se genera a través de una secuencia automática.*

*\* Con referencia a la operación sería un CHECK para ( D=Delete, I=Insert, U=Update).  
Tabla corresponde a la  
tabla afectada.*

*\*/*

*create table auditoria (  
aud\_registro number primary key,  
aud\_tabla varchar2(30),  
aud\_operacion varchar2(1),  
aud\_usuario varchar2(30),*

```
aud_fecha date,  
constraint operacion_c check(aud_operacion in('D','I','U'))  
);
```

```
create sequence sec_auditoria start with 1 increment by 1;
```

```
/*  
  
Tipocliente (tip_código number(2), tip_descripción varchar2(30). La llave primaria es código.  
(  
1=Socio, 2= Ahorrista, 3 = tercero) */
```

```
create table TipoCliente (  
tip_codigo number(2) not null primary key,  
tip_descripcion varchar2(30),  
constraint tip_codigo_c check (tip_codigo >= 1 and tip_codigo <= 3)  
);
```

```
/*  
  
Cliente ( cli_numcliente number (8), cli_tipo number(2) CHECK(1,2,3) cli_cedula  
varchar2(25),  
cli_nombre varchar2(25), cli_apellido varchar2(25), cli_sexo varchar2(1) CHECK (F,M),  
cli_fecha_Nacimiento date)
```

\* (llave primaria cli\_numcliente es un número que se genera de manera automático por medio de un secuenciador

que deberá crearse previamente. La llave foránea tipo con la tabla tip\_tipocliente.

\*/

create sequence sec\_cliente start with 1 increment by 1;

create table cliente (

cli\_numcliente number(8) not null primary key,

cli\_tipo number(2),

cli\_cedula varchar2(25),

cli\_nombre varchar2(25),

cli\_apellido varchar2(25),

cli\_sexo varchar2(1),

cli\_fecha\_nacimiento date,

constraint cli\_sexo\_c check (cli\_sexo in('F','M')),

constraint cli\_tipo\_c check (cli\_tipo >= 1 and cli\_tipo <= 3),

constraint cliente\_fk foreign key (cli\_tipo) references tipocliente (tip\_codigo)

);

/\*

Tipoahorro ( taho\_tipo number(2), taho\_descripción varchar2 (30). La llave primaria es tipo.

(

1=Navidad, 2=corriente, 3=escolar) \*/

create table tipoahorro (

taho\_tipo number(2) not null primary key,

taho\_descripcion varchar2(30),

constraint taho\_tipo\_c check (taho\_tipo >= 1 and taho\_tipo <= 3)

);

/\*

Ahorros ( aho\_numcliente number(8), aho\_tipoahorro number(2) CHECK(1,2,3),  
aho\_cuenta number(8) es un secuenciador,  
aho\_cuota number(15,2), aho\_Saldo number(15,2), aho\_ahomensual number(15,  
2), aho\_retmensual(15,2), fecha date, usuario).

(llave primaria (aho\_numcliente, aho\_tipoahorro, aho\_cuenta). (llave foránea aho\_tipoahorro  
versus tabla de  
tipoahorro, aho\_numcliente versus cliente.

\*/

create sequence sec\_aho\_cuenta start with 1 increment by 1;

create table ahorros (  
aho\_numcliente number(8),  
aho\_tipoahorro number(2),  
constraint aho\_tipoahorro\_c check (aho\_tipoahorro >= 1 and aho\_tipoahorro <= 3),  
aho\_cuenta number(8),  
aho\_cuota number(15,2),  
aho\_saldo number(15,2),  
aho\_ahomensual number(15,2),  
aho\_retmensual number(15,2),  
fecha date,  
usuario varchar2(40)

```
);  
/  
/*
```

Procesos básicos.

Todos los objetos de base de datos ya fueron creados y tablas Tipocliente, Cliente, tipoahorro cuentan con la información que corresponde, cargada.

### REALICE LAS SIGUIENTE TAREAS PARA LA BASE DE DATOS DEL BANCO

Levantar los siguientes los objetos como programas almacenados de bases de datos

\* Una secuencia para usar como incremento automático para la tabla de auditoria en el número de registro.

```
*/
```

-- ya fue creada dicha secuencia.

```
/*
```

\* Un Disparador (insert, update, delete) para la tabla de auditoria cuando se crean los registros y se actualiza información en la tabla ahorros. 10pts

```
*/
```

```
create or replace trigger tr_auditoria  
after insert or update or delete on ahorros  
for each row  
declare  
v_operacion varchar2(1);
```

```
begin
```

```
if inserting then
```

```
v_operacion = 'I';
```

```
insert into auditoria values(sec_auditoria.nextval,'Tabla ahorros',v_operacion,user,sysdate);
```

```
elsif updating then
```

```
v_operacion = 'U';
```

```
insert into auditoria values(sec_auditoria.nextval,'Tabla ahorros',v_operacion,user,sysdate);
```

```
elsif deleting then
```

```
v_operacion = 'D';
```

```
insert into auditoria values(sec_auditoria.nextval,'Tabla ahorros',v_operacion,user,sysdate);
```

```
end if;
```

```
end tr_auditoria;
```

```
/
```

```
/*
```

*\* Un Procedimiento para la creación de los registros iniciales en la tabla de ahorros, los campos de monto en la tabla serán cargados en cero, los tipos de ahorros serán elegidos por los estudiantes en la invocación con los parámetros. Agregar Excepciones. 5 pts*

```
*/
```

```
create or replace procedure pr_reg_ahorros (
```

```
p_aho_numcliente ahorros.aho_numcliente%type,
```

```
p_aho_tipoahorro ahorros.aho_tipoahorro%type,
```

```

--p_aho_cuenta ahorros.aho_cuenta%type,
p_aho_cuota ahorros.aho_cuota%type,
p_aho_saldo ahorros.aho_saldo%type,
p_aho_ahomensual ahorros.aho_ahomensual%type,
p_aho_retmensual ahorros.aho_retmensual%type,
p_fecha ahorros.fecha%type,
p_usuario ahorros.usuario%type
)
as
begin
insert into ahorros values
(p_aho_numcliente,p_aho_tipoahorro,sec_aho_cuenta.nextval,p_aho_cuota,p_aho_saldo,p_aho_
homensual,
p_aho_retmensual,p_fecha,p_usuario);
exception
when dup_val_on_index then
dbms_output.put_line('Datos repetidos');
when value_error then
dbms_output.put_line('Error en el tamaño de los datos ingresados');
when others then
dbms_output.put_line('Error en la inserción de los datos');
end pr_reg_ahorros;
/
begin
pr_reg_ahorros(1,1,1,600.00,300.00, 200.00, sysdate,user);
end;
/

```

/\*

\* Un Procedimiento almacenado para los depósitos en los ahorros (afecta los campos de aho\_saldo, aho\_ahomensual, aho\_fecha, aho\_usuario) Agregar Excepciones 5 ptos

\*/

```
create or replace procedure pr_deposito_ahorros (  
  p_aho_ahomensual ahorros.aho_ahomensual%type,  
  p_fecha ahorros.fecha%type,  
  p_usuario ahorros.usuario%type  
)  
as  
  cursor c_cursor is select  
    aho_numcliente, aho_tipoahorro, aho_cuenta, aho_cuota, aho_saldo, aho_ahomensual, aho_retmens  
al, fecha, usuario from ahorros;  
begin  
  
  for v_cursor in c_cursor loop  
    update ahorros  
    set aho_saldo = aho_saldo + p_aho_ahomensual,  
        aho_ahomensual = aho_ahomensual + p_aho_ahomensual,  
        fecha = p_fecha,  
        usuario = p_usuario;  
  end loop;  
  
exception  
  when dup_val_on_index then
```



```

dbms_output.put_line('Datos repetidos');
when value_error then
dbms_output.put_line('Error en el tamaño de los datos ingresados');
when others then
dbms_output.put_line('Error en la inserción de los datos');
end pr_deposito_ahorros;
/
/*
* Un Procedimiento almacenado para los retiros (afecta los campos de aho_saldo,
aho_retmensual, aho_fecha, aho_usuario). Agregar Excepciones 5 ptos
*/
create or replace procedure pr_ret (
p_aho_retmensual ahorros.aho_retmensual%type,
p_fecha ahorros.fecha%type,
p_usuario ahorros.usuario%type
)
as
cursor c_cursor is select
aho_numcliente, aho_tipoahorro, aho_cuenta, aho_cuota, aho_saldo, aho_ahomensual, aho_retmensual,
fecha, usuario from ahorros;
begin

for v_cursor in c_cursor loop
update ahorros
set aho_saldo = aho_saldo - p_aho_retmensual,
aho_retmensual = aho_retmensual + p_aho_retmensual,
fecha = p_fecha,

```

```
usuario = p_usuario;
```

```
end loop;
```

```
exception
```

```
when dup_val_on_index then
```

```
dbms_output.put_line('Datos repetidos');
```

```
when value_error then
```

```
dbms_output.put_line('Error en el tamaño de los datos ingresados');
```

```
when others then
```

```
dbms_output.put_line('Error en la inserción de los datos');
```

```
end pr_ret;
```

```
/
```

```
/*
```

*\* Un Función que calcula el interés de los ahorros corriente (saldo \* 4% \* 30 días) \$65. Este proceso actualiza el saldo y el ahorro mensual de tabla de ahorros. Este proceso es una vez culmina el mes. 5 ptos*

```
*/
```

```
create or replace function fn_calc_int (
```

```
p_aho_saldo ahorros.aho_saldo%type
```

```
)
```

```
return number
```

```
as
```

```
interes number;
```

```
begin
```

```
interes = (p_aho_saldo * 0.04 * 30 )/65;
```

```
return interes;  
end fn_calc_int;  
/
```

```
alter table ahorros add (saldo1 number, saldo2 number);
```

```
-- bloque anónimo
```

```
declare
```

```
v_saldo ahorros.aho_saldo%type;
```

```
v_ahomensual ahorros.aho_ahomensual%type;
```

```
v_aho_retmensual ahorros.aho_retmensual%type;
```

```
cursor c_cursor is select aho_numcliente, aho_tipoahorro aho_cuenta, aho_cuota  
aho_saldo, aho_ahomensual, aho_retmensual, fecha, usuario from ahorros;
```

```
begin
```

```
for v_cursor in c_cursor loop
```

```
v_saldo = v_cursor.aho_saldo;
```

```
v_ahomensual = v_cursor.aho_ahomensual;
```

```
v_aho_retmensual = v_cursor.aho_retmensual;
```

```
-- para deposito y retiro en la tabla ahorros
```

```
update ahorros
```

```
set aho_saldo = v_saldo,
```

```
aho_ahomensual = v_ahomensual,
```

```
saldo1 = v_saldo + v_ahomensual,
```

```
saldo2 = v_saldo - v_aho_retmensual;
```

```
end loop;
```

```

exception
when dup_val_on_index then
dbms_output.put_line('Datos repetidos');
when value_error then
dbms_output.put_line('Error en el tamaño de los datos ingresados');
when others then
dbms_output.put_line('Error en la inserción de los datos');
end;
/
/*
* Crear una vista sobre la tabla de ahorros. 5 ptos

```

*Nota: Recuerde realizar las invocaciones con los parámetros correctos.*

```

*/

-- para deposito
create or replace view vista1 as select aho_saldo as saldo, aho_ahomensual as monto, saldo1
from ahorros;
select * from vista1;

-- para retiro
create or replace view vista2 as select aho_saldo as saldo, aho_retmensual as monto, saldo2
from ahorros;
select * from vista2;

```