

UNIVERSIDAD TECNOLÓGICA DE PANAMA
FACULTAD DE INGENIERIA DE SISTEMAS COMPUTACIONALES
LICENCIATURA EN INGENIERIA DE SISTEMAS DE INFORMACION
SISTEMAS DE BASE DE DATOS II

25
38

PARCIAL NO.3

NOMBRE: Jenny B FECHA: 16/11/2022
CEDULA: Michael Solis GRUPO: 11F131

El Banco Nacional de Mis Provincias, desea 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.

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

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.

Tipoahorro (taho_tipo number(2), taho_descripción varchar2 (30)). La llave primaria es tipo. (1=Navidad, 2=corriente, 3=escolar)

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). sec-aho

- Llave primaria (aho_numcliente, aho_tipoahorro, aho_cuenta). Llave foránea aho_tipoahorro versus tabla de tipoahorro, aho_numcliente versus cliente.

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.

GOOD LUCKY

- **Un Disparador** (insert, update, delete) para la tabla de auditoria cuando se crean los registros y se actualiza información en la tabla ahorros. 10 pts
- **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
- **Un Procedimiento** almacenado para los depósitos en los ahorros (afecta los campos de aho_saldo, aho_ahomensual, aho_fecha, aho_usuario) . Agregar Excepciones 5 pts
- **Un Procedimiento** almacenado para los retiros (afecta los campos de aho_saldo, aho_retmensual, aho_fecha, aho_usuario). Agregar Excepciones 5 pts
- **Un Función** que calcula el interés de los ahorros corriente ($\text{saldo} * 4\% * 30 \text{ días} / 365$). Este proceso actualiza el saldo y el ahorro mensual de tabla de ahorros. Este proceso es una vez culmina el mes. 5 pts
- Crear una vista sobre la tabla de ahorros. 5 pts

si es

SALDO		
deposito		
saldo	monto	saldo
500	100	600

→

MENSUAL		
deposito		
ahorromen	monto	ahorromen
100	100	200

si es

SALDO		
retiro		
saldo	monto	saldo
500	100	400

→

MENSUAL		
retiro		
retiromen	monto	retiromen
100	100	200

Nota: recuerde realizar las invocaciones con los parámetros correctos

Cursor sobre la tabla que actualice

create or replace trigger tr_crear-actualizar
 After insert or delete or update or alter
 for each row

declare

V-letra varchar2(1);

begin

if inserting then

V-letra := 'I';

insert into auditoria (and_registro, and_tabla, and_operacion,
 and_usuario, and_fecha)

Values (and_registro.NextVal, :New.aho-tiposahorro, V-letra,
 :New.usuario, :New.fecha);

end if;

if updating then

V-letra := 'U';

insert into auditoria (and_registro, and_tabla, and_operacion,
 and_usuario, and_fecha)

Values (and_registro.nextval, :New.aho-tiposahorro, V-letra,
 :New.usuario, :New.fecha);

End if;

if deleting then

V-letra := 'D';

insert into auditoria (and_registro, and_tabla, and_operacion,
 and_usuario, and_fecha)

Values (and_registro.nextval, :New.aho-tiposahorro, V-letra,
 :New.usuario, :New.fecha);

end if;

exception

when dup-val-on-index then

dbms_output.put_line ('Datos repetidos');

when others then dbms_output.put_line ('Error en la inserción');

end tr_crear-actualizar;

or replace procedure Pr-dif-ahorros(
 aho-saldo IN ahonos. aho-saldo%+YPE,
 P-aho-ahomensual ahonos. aho-mensual%+YPE,
 P-aho-fecha ahonos. aho-fecha%+YPE,
 P-aho-usuario ahonos. aho-usuario%+YPE)

begin
 as

insert into ahonos (aho-saldo, aho-ahomensual,
 aho-fecha, aho-usuario) values (P-aho-saldo,
 P-aho-ahomensual, P-aho-fecha, P-aho-usuario);

exception

When dup-val-or-index then
 dbms_output.put_line ('Datos repetidos');

When others then
 dbms_output.put_line ('Error en la inserción
 de datos');

end Pr-dif-ahonos;

Create or replace procedure Pr-retiros(
 P-aho-saldo ahonos. aho-saldo%+YPE,
 P-aho-retmensual ahonos. aho-retmensual%,
 P-aho-fecha ahonos. aho-fecha%+YPE,
 P-aho-usuario ahonos. aho-usuario%+YPE)

or

begin

insert into ahonos values (P-aho-saldo, P-aho-retmensual,
 P-aho-fecha, P-aho-usuario);

exception

When dup-val-or-index then
 dbms_output.put_line ('Datos repetidos');

When others then dbms_output.put_line ('Error en la inserción');

end Pr-retiros;


```

Create sequence sec-auditoria
start With 1 increment by 1;
Create sequence sec-aho-cuenta start with 1 increment by 1;
Create or replace procedure pr-reg-ahorros
(
p-aho-numcliente ahores.aho-numcliente%TYPE,
p-aho-tipoahorro ahores.aho-tipoahorro%TYPE,
p-aho-cuenta ahores.aho-cuenta%TYPE,
p-aho-saldo ahores.aho-saldo%TYPE,
p-aho-atomensual ahores.aho-atomensual%TYPE,
p-aho-retmensual ahores.aho-retmensual%TYPE,
p-fecha ahores.fecha%TYPE, p-usuario ahores.usuarios%TYPE)
as
begin
insert into ahores values (p-aho-numcliente, p-aho-
tipoahorro, sec-aho-cuenta.NextVal, p-aho-cuenta, p-aho-
saldo, p-aho-atomensual, p-aho-retmensual, p-fecha, p-usuario);
exception
When dup-val-on-index then
dems-output.put-line('Datos repetidos');
When others then
dems-output.put-line('Error en la inserción de datos');
end pr-reg-ahorros;

```

Batista, Jerry
 Davis, Michael

get or replace function fn-calc

P-aho-saldo ahorros.aho-saldo%TYPE)

return number

as

V-total number;

begin

V-total := (P-aho-saldo * 0.04 * 30) / 365;

return V-total;

end fn-calc;

Create or replace View tabla-ahorros as select a.aho-numcliente,
a.aho-saldo, a.aho-ahomensual, a.aho-retmensual,

c.numcliente as ahorros a

inner join cliente c

On a.aho-numcliente = c.cl-numcliente;

-- Bloque anónimo

declare

V-tipo tipoahorro. taho-tipo%TYPE;

V-aho-saldo ahorros. aho-saldo%TYPE;

V-aho-mensual ahorros. aho-mensual%TYPE;

Cursor C-cursor is select aho-saldo from ahorros;

open C-cursor;

begin

loop

fetch C-cursor into V-aho-saldo

exit when C-cursor % not found;

update ahorros

set aho-saldo = V-aho-saldo;

end loop;

exception

Proforma
cheque

Have

When dup- val- or- index then
dbms- output. put- line ('Datos repetidos');
When other then
dbms- output. put- line ('Error en la inserción de
datos');
end;

Balistas, Jimmy
Dolis, Michael