Parcial #3 Victor Rodriguez

```
· Create Sequence aud_registro_seq Start with |20-70-7414
1   1 INCREMENT By 1;

CREATE OR REPLACE TRIGGER t_aud_registro_ins
AFTER INSERT ON Ahorros
FOR EACH ROW
 BEGIN
 INSERT into Auditoria( aud_registro, aud.tabla, ~~aud_op~~
 aud_operacion, aud_usuario, aud_fecha, saldo_inicial,
mon_toaplicar, saldo_final, usuario)
 VALUES (aud_registro_seq.nextval, 'Ahorros', 'I', :USER
 SYSTIMESTAMP, NULL, :NEW.aho_cuota, :NEW.aho_saldo, :New:usuario
 END t_aud_registro_ins; /
CREATE OR REPLACE TRIGGER t_aud_registro_upd.
AFTER UPDATE ON Ahorros
.FOR EACH ROW
  BEGIN
  INSERT into Auditoria(aud_registro, aud_tabla, aud_opc,
  aud_usuario, aud_fecha, saldo_inicial, monto aplicar, saldo_
  final, usuario) VALUES (aud_registro_seq.NEXTVAL, 'Ahorros',
  'U', USER, SYSTIMESTAMP, :OLD.aho_saldo, :NEW.aho_cuota,
  :NEW.aho_saldo ;NEW.usuario);
  END t_aud_registro_UPD; /

  CREATE OR REPLACE TRIGGER t_aud_registro_del
  AFTER DELETE ON Ahorros
  FOR EACH ROW   BEGIN
  INSERT into auditoria (aud_registro, aud, tabla, aud_operacion
aud_usuario, aud-fecha, saldoinicial, MontoAplicar, saldofinal, Usuario)
VALUES (aud_registro-seq. NEXTVAL, 'Ahorros', 'D', USER,
SYSTIMESTAMP, :OLD.aho-saldo, :NEW.aho-cuota, :old.aho-saldo
:NEW. USER);  END t-aud-registro_del;/
CREATE Sequence Aho-cuenta-seq Start with 1 increment by 1;
CREATE OR REPLACE PROCEDURE insertar Ahorros (
P_aho-Numcliente IN Ahorros.aho_Numcliente %TYPE,
P_aho-tipoahorro IN Ahorros.aho-tipoAhorro %TYPE,
P-Check  IN Ahorros.P-check %TYPE,
P-aho-cuota IN Ahorros.aho-cuota %TYPE :=0,
P_aho-Saldo  IN Ahorros. aho-saldo %TYPE :=0,
P-ahomensual IN Ahorros. ahoMensual %TYPE :=0,
P-aho-retmensual IN Ahorros . aho-retmensual %TYPE:=0,
P-usuario IN Ahorros. usuario %TYPE
AS
BEGIN
INSERT INTO Ahorros ( aho-Num-cliente, tipo-ahorro, check
aho-cuenta, aho_cuota, aho_Saldo -aho-ahomensual, ~~aho-~~
aho-retmensual, fecha, usuario) VALUES (
P_aho-Numcliente, P_aho-tipoAhorro, P-cKeck,
aho-cuenta-seq.Nextval, P-aho-cuota, P-aho-saldo, P-aho-ahoMensual,
P_aho-retmensual, SYSTIMESTAMP, P-Usuario)
EXCEPTION
    WHEN DUP_VAL-ON_INDEX tHEN
    DBMS-OUTPUT.Put-LINE('ERROR. Ya existe la cuenta');
  WHEN others then
    DBMS_output.put-Line ('error insertando la cuenta); Commit
END insertar Ahorros; /
```

UR 2070

```
CREATE OR REPLACE PROCEDURE depositar A (
P_aho_saldo IN Ahorros.aho_saldo %type,
P_aho_mensual IN Ahorros.aho_mensual %TYPE,
P_aho_usuario IN Ahorros.aho_usuario %type.
P_aho_cuenta IN Ahorros.aho_aho_cuenta %type)AS
BEGIN
  UPDATE Ahorros
     SET aho_saldo := P_aho_saldo;
         aho_mensual := P_aho_mensual, fecha := sys timestamp,
         aho_usuario := P_aho_usuario,
    where Aho_cuenta = P_aho_cuenta;
EXCEPTION
   When NO_DATA_FOUND then
       DBMS_OUTPUT.Put_Line ('cuenta no encontrada');
   When others then
       DBMS_output.putline ('error actualizando la tabla
                                        ahorros');
END depositarA;/

CREATE OR REPLACE PROCEDURE retirar (
P_aho_saldo, IN Ahorros.aho_saldo %type,
P_aho_retmensual IN Ahorros_aho_retmensual %type,
P..aho_usuario IN Ahorros.~~uso~~aho_usuario %type.
P_aho_cuenta IN Ahorros.aho_cuenta %type,
P_Num_cliente IN Ahorros Num Cliente %type,
P_aho_tipoAho IN Ahorros.aho.tipo Aho %type) AS
BEGIN UPdate Ahorros
  SET aho_saldo := P_aho saldo, aho_retmensual := 
  P_aho_retmensual, aho_usuario := P.aho.usuario,
 where aho_cuenta = P.aho_cuenta AND numcliente = P.Num.cliente
  AND aho.tipoAho = P_aho_tipo Aho;)
```

```
EXCEPTION
  WHEN NO_DATA_FOUND then
    DBMS_____ ('no se encontró el usuario');
  when others then
    DBMS______ ('error al retirar de la cuenta');
END retirar;/

CREATE OR REPLACE FUNCTION calcularinteres(
 P_aho_numclienteIN Ahorros.aho_numcliente%type
P_aho_tipoAhorro IN Ahorros.aho_tipoAho%type;
P_aho_cuenta IN Ahorros.aho_cuenta%type;)AS
V_saldo Ahorros.aho_saldo%type;
 V_interes Number;
Begin
 Select aho_saldo INTO V_saldo FROM Ahorros
  where aho_numcliente = P_aho_numcliente
          aho_tipoAhorro = P__aho tipoAhorro,
          aho_cuenta = P_aho_cuenta;
V_interes:=(V_saldo *0.04*30)/365;

 Update Ahorros
      Set aho_saldo = V_saldo + V_interes,
          aho_mensual = Vinteres;
    where aho numcliente= P_aho+numcliente,
          aho_tipo ahorro - P_ tipoAhorro,
          aho_cuenta = P_aho_cuenta;

EXCEPTION WEN NODADA_FOUND then
```

VR 20 70

```
When NO-DATA_FOUND then
  DBMS_OUTPU ... ('No se encontro el cliente');
when others then
  DBMS_output_... ('error al calcular el interes');
END calcularInteres; /

CREATE OR REPLACE VIEW vista_ahorros AS
  V-Num cliente %AHORROS.aho-Num cliente %type
  Select aho-NumCliente,
         aho-cuenta;
         aho-saldo,
         aho_cuota
  FROM ahorros
  where aho-NumCliente = V-Num_cliente ;
```

VR 20-20