

Alunos: Luís Miguel Torres R. e Guilherme Toffoli Bedin

– Script

```
CREATE TABLE agencia (codag number(6) primary key, nomeag varchar(20));
```

```
CREATE TABLE cliente (codigo number(4) PRIMARY KEY, nome  
varchar(40), cidade varchar(20), uf char(2));
```

```
CREATE TABLE conta (numero number(10) PRIMARY KEY, saldo  
number(20,4), codag number(6), limite number (20,4), constraint  
fk_agencia foreign key(codag) references agencia(codag));
```

```
CREATE TABLE movimento (codmov number(20) PRIMARY KEY, tipo  
Char(1), datahora timestamp, numero number(10), valor  
number(20,4), FOREIGN KEY(numero) REFERENCES conta (numero));
```

```
CREATE TABLE cliente_conta (numero number(10), codigo  
number(4),propriedade varchar(20),  
constraint fk_conta FOREIGN KEY(numero) REFERENCES conta  
(numero),  
constraint fk_cliente FOREIGN KEY(codigo) REFERENCES cliente  
(codigo));
```

-- Questão 2

```
insert into agencia values (1, 'Agencia 1');  
insert into agencia values (2, 'Agencia 2');  
insert into agencia values (3, 'Agencia 3');  
insert into agencia values (4, 'Agencia 4');
```

```
insert into cliente (codigo, nome, cidade, uf) values (1, 'Luís M.', 'Chapecó', 'SC');  
insert into cliente (codigo, nome, cidade, uf) values (2, 'Pedro R.', 'Chapecó', 'SC');  
insert into cliente (codigo, nome, cidade, uf) values (3, 'Adriana R.', 'Campos Novos', 'SC');  
insert into cliente (codigo, nome, cidade, uf) values (4, 'Rafael T.', 'São Paulo', 'SP');
```

```
insert into conta values(1111111111, 20000, 1, 5000);  
insert into conta values(2222222222, 25000, 1, 15000);  
insert into conta values(3333333333, 30000, 2, 1000);  
insert into conta values(4444444444, 5000, 3, 1000);
```

```
insert into movimento values(10, 1, '02-04-2021 10:40:00', 1111111111, 500);  
insert into movimento values(20, 1, '10/12/2022 12:30:00', 2222222222, 1000);  
insert into movimento values(30, 0, '20/01/2022 09:21:00', 3333333333, 2500);  
insert into movimento values(40, 0, '30/05/2021 08:25:00', 4444444444, 5000);
```

```
insert into cliente_conta values(1111111111, 1, 'Titular');  
insert into cliente_conta values(2222222222, 2, 'Não Titular');  
insert into cliente_conta values(3333333333, 3, 'Não Titular');  
insert into cliente_conta values(4444444444, 4, 'Titular');
```

-- Questão 3

```
create or replace trigger saldo_atualizado
after insert on movimento
for each row
declare
    saldo_conta number(20,4);
begin
    select saldo into saldo_conta
    from conta
    where conta.numero = :NEW.numero;

    if :NEW.valor <= saldo_conta then
        saldo_conta := saldo_conta - :NEW.valor;
        update conta set saldo = saldo_conta where conta.numero = :NEW.numero;
    else
        dbms_output.put_line('Saldo insuficiente');
    end if;
end;
```

– Questão 4

```
CREATE OR REPLACE FUNCTION saldo_medio_conta(
    numero_conta IN NUMBER,
    data_inicio IN TIMESTAMP,
    data_fim IN TIMESTAMP
) RETURN NUMBER
IS
    saldo_medio NUMBER;
BEGIN
    SELECT AVG(valor)
    INTO saldo_medio
    FROM movimento
    WHERE numero = numero_conta
    AND datahora BETWEEN data_inicio AND data_fim;

    RETURN saldo_medio;
END saldo_medio_conta;
```

– Questão 5

```
create view titular_nome as
select
    c.nome as nome,
    ct.saldo as saldo
from
    cliente c
join
    cliente_conta co on co.codigo = c.codigo
join
```

```
conta ct on co.numero = ct.numero  
where co.propriedade = 'Titular';
```

– Questão 6

```
CREATE OR REPLACE PROCEDURE manutencao(valor_manutencao IN NUMBER)  
is  
begin  
    update conta  
    set saldo = saldo - valor_manutencao;  
end;
```

– Questão 7

```
CREATE OR REPLACE FUNCTION valor_total_deposito(data_inicio IN TIMESTAMP,  
data_fim IN TIMESTAMP, numero_conta IN NUMBER)  
RETURN NUMBER  
IS  
    valor_total NUMBER;  
BEGIN  
    SELECT SUM(valor)  
    INTO valor_total  
    FROM movimento  
    WHERE numero = numero_conta  
    AND datahora BETWEEN data_inicio AND data_fim;  
  
    RETURN valor_total;  
END valor_total_deposito;
```

– Questão 8

```
create or replace package todas_funcoes as  
    FUNCTION saldo_medio_conta(numero_conta IN NUMBER,data_inicio IN  
TIMESTAMP,data_fim IN TIMESTAMP) RETURN NUMBER;  
    PROCEDURE manutencao(valor_manutencao IN NUMBER);  
    FUNCTION valor_total_deposito(data_inicio IN TIMESTAMP, data_fim IN TIMESTAMP,  
numero_conta IN NUMBER) RETURN NUMBER;  
end todas_funcoes;
```

```
create or replace package body todas_funcoes as
```

```
    FUNCTION saldo_medio_conta(  
        numero_conta IN NUMBER,  
        data_inicio IN TIMESTAMP,  
        data_fim IN TIMESTAMP  
    ) RETURN NUMBER  
    IS  
        saldo_medio NUMBER;
```

```

BEGIN
    SELECT AVG(valor)
    INTO saldo_medio
    FROM movimento
    WHERE numero = numero_conta
    AND datahora BETWEEN data_inicio AND data_fim;

    RETURN saldo_medio;
END saldo_medio_conta;

PROCEDURE manutencao(valor_manutencao IN NUMBER)
is
begin
    update conta
    set saldo = saldo - valor_manutencao;
end;

FUNCTION valor_total_deposito(data_inicio IN TIMESTAMP, data_fim IN TIMESTAMP,
numero_conta IN NUMBER) RETURN NUMBER
IS
    valor_total NUMBER;
BEGIN
    SELECT SUM(valor)
    INTO valor_total
    FROM movimento
    WHERE numero = numero_conta
    AND datahora BETWEEN data_inicio AND data_fim;

    RETURN valor_total;
END valor_total_deposito;
end todas_funcoes;

```