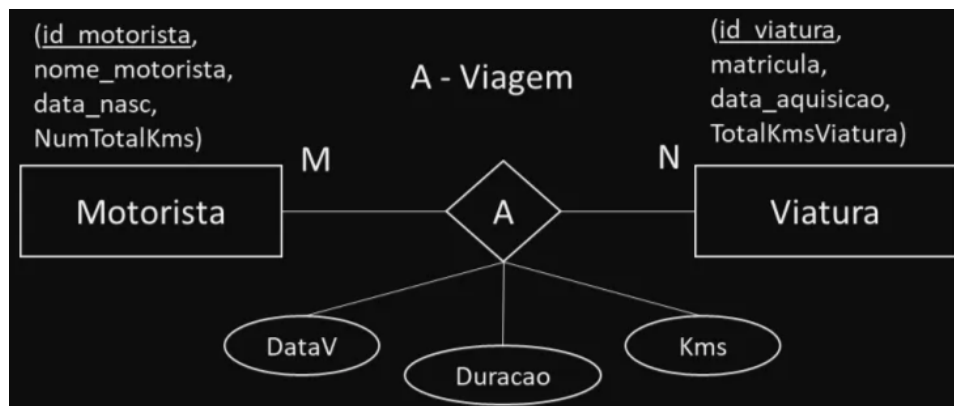




# Frequência Modelo

## Grupo I

1. Perante o seguinte modelo entidade e relacionamento, de uma organização que guarda registo dos seus motoristas e das viaturas usadas nas viagens que realizam:



a) Um procedimento (ListaViagensMotorista) que receba como parâmetro de entrada um identificador de um motorista (id\_motorista), e liste todos os dados referentes às viagens por ele realizadas, tendo por base o exemplo seguinte:

Considere o modelo apresentado, e com recurso a PL/SQL apresente:

**EX:**

**Motorista: Rui Miguel — Data Nasc: 2000.01.01 — NumTotalKms: 120**

**Viagem1: DataV: 2022.01.01 — Duracao: 25 — Kms: 10 — Matricula: AA-01-AA**

**Viagem2: DataV: 2022.01.02 — Duracao: 75 — Kms: 80 — Matricula: AA-02-AA**

**Viagem1: DataV: 2022.01.01 — Duracao: 25 — Kms: 10 — Matricula: AA-01-AA**

...

```
CREATE OR REPLACE PROCEDURE ListaViagensMotorista (vp_id_moto
IS

v_regMotorista motorista%ROWTYPE;
v_matricula viatura.matricula%TYPE;

CURSOR c_viagens IS
    SELECT * FROM viagem WHERE id_motorista = vp_id_motorista

BEGIN
    SELECT *
    INTO v_regMotorista
    FROM motorista
    WHERE id_motorista = vp_id_motorista;

    DBMS_OUTPUT.PUT_LINE('Motorista : ' || v_regMotorista.nom
                        || ' -- Data Nasc : '
                        || ' -- Kms: ' || v_r

    FOR v_regViagem IN c_viagens LOOP
        SELECT matricula
        FROM v_matricula
        FROM viatura
        WHERE id_viatura = v_regViagem.id_viatura;

        DBMS_OUTPUT.PUT_LINE(' ----- Viagem: ' || c_v
```

```

|| ' -- DataV: ' || v
|| ' -- Duracao: ' ||
|| ' -- Kms: ' || v_r
|| ' Matricula: ' ||

END LOOP;
END;

```

**b) Uma função (DataPrimeiraViagem) que receba como parâmetro de entrada um identificador de um motorista (id\_motorista), e devolva a data da primeira viagem por ele realizada.**

```

CREATE OR REPLACE FUNCTION DataPrimeiraViagem
(vf_id_motorista IN motorista.id_motorista%TYPE) RETURN DATE

IS

v_dataV viagem.datav%TYPE;

BEGIN
    SELECT MIN(datav)
    INTO v_dataV
    FROM viagem
    WHERE id_motorista = vf_id_motorista;

    RETURN(v_dataV);
END;

```

**c) Um *trigger* que por cada nova viagem atualize, em coerência, os atributos 'NumTotalKms' e 'TotalKmsViatura'.**

```

CREATE OR REPLACE TRIGGER atualiza_kilometros
AFTER INSERT ON viagem
FOR EACH ROW

BEGIN
    IF INSERTING THEN
        UPDATE motorista
        SET numTotalKms = numTotalKms + :NEW.kms
        WHERE id_motorista = :NEW.id_motorista;

        UPDATE viatura
        SET totalKmsViatura = totalKmsViatura + :NEW.kms
        WHERE id_viatura = :NEW.id_viatura;
    END IF;
END;

```

## 2. Considere ainda o esquema anterior. Escreva:

a) Um trigger que aquando de uma nova inserção ou atualização, verifique as seguintes restrições:

**INSERÇÃO / ACTUALIZAÇÃO:** o nome do motorista deve ser armazenado em letras maiúsculas

```

CREATE OR REPLACE TRIGGER nome_maiusculo
BEFORE INSERT OR UPDATE ON motorista
FOR EACH ROW

BEGIN
    NEW.nome_motorista := UPPER(:NEW.nome_motorista);
END;

```

**b) Crie em PL/SQL uma função (geraData) que dados dois anos, retorne uma data criada aleatoriamente e que esteja compreendida entre os anos indicados. Deve ter em conta a seguinte informação:**

Dias	Meses
31	Janeiro, Março, Maio, Julho, Agosto, Outubro, Dezembro
30	Abril, Junho, Setembro, Novembro
28	Fevereiro
29	Fevereiro (se ano bissexto)

```
CREATE OR REPLACE FUNCTION geraData (ano1 IN INT, ano2 IN INT)
IS
    tmpAno INT;
    tmpMes INT;
    tmpDia INT;
    tmpString VARCHAR(20)

BEGIN
    IF ano1 > ano2 THEN
        tmpAno := Round(dbms_random.value(ano2, ano1), 0);
    ELSE
        tmpAno := Round(dbms_random.value(ano1, ano2), 0);
    END IF;

    tmpMes := Round(dbms_random.value(1, 12), 0);

    IF tmpMes = 2 THEN
        IF (Mod(tmpAno, 4) AND Mod(tmpAno, 100) OR Mod(tmpAno, 400) = 0) THEN
            tmpDia := Round(dbms_random.value(1, 29), 0);
        ELSE
            tmpDia := Round(dbms_random.value(1, 28), 0);
        END IF;
    ELSIF tmpMes = 4 OR tmpMes = 6 OR tmpMes = 9 OR tmpMes = 11
```

```
        tmpDia := Round(dbms_random.value(1, 30), 0);  
ELSE  
        tmpDia := Round(dbms_random.value(1, 31), 0);  
END IF;  
  
tmpString := tmpDia || '-' || tmpMes || '-' || tmpAno;  
return to_date(tmpString, 'DD/MM/YYYY');  
END;
```

## Grupo II

### 1. O seguinte texto reflete as preocupações de um decisor numa organização:

"A empresa EST\_TRAVEL, de uma operadora de viagens às capitais europeias, em minibus (exclusivamente de 9 lugares), necessita de obter respostas às seguintes questões:

1. Qual a distribuição, por idade, que solicita mais viagens;
2. Dos destinos existentes, qual o mais lucrativo;
3. Qual o número médio de passageiros, por viagem, aos fins-de-semana;
4. Qual a distribuição, do número de viagens mensalmente, por capital..."

Tendo em conta o texto, e os conceitos de sistema de suporte à decisão:

**a) Apresente, de forma justificada, uma possível estrutura para a tabela de factos.**

### Dim\_Tempo

Tempo\_ID,  
Ano,  
Mes,  
Dia,  
Flag\_Dia\_Semana

### Dim\_Cliente

Cliente\_ID  
Nome\_Cliente  
Data\_Nascimento

### TB\_FACTOS

Cliente\_ID  
Destino\_ID  
Tempo\_ID  
Viagem\_ID

Custo/Lucro  
#Passageiros  
#Viagens

### Dim\_Viagem

Viagem\_ID  
Nome\_Viagem

### Dim\_Destino

Destino\_ID  
Nome\_Destino