

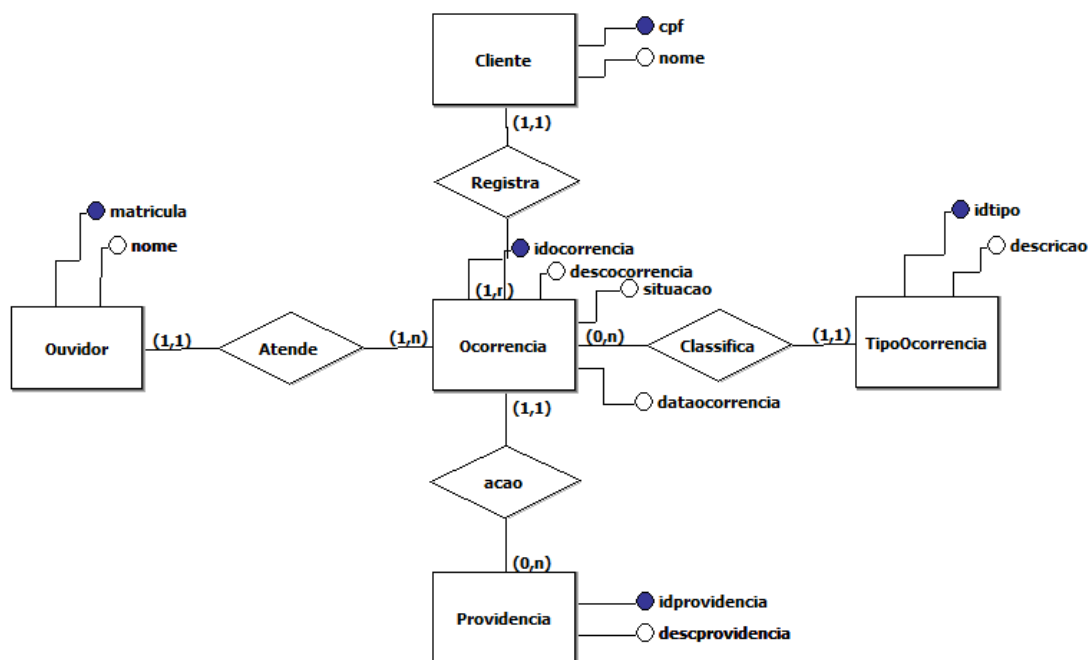
1ª AVALIAÇÃO

ALUNO	Diogo Lourenço da Costa Felipe			MATRÍCULA	04050521
DISCIPLINA	PROJETO DE BANCO DE DADOS			DATA DA PROVA	29/04/2020
PROFESSOR	ALMIR MONTEIRO JUNIOR			TIPO DE PROVA	UNICA
TURMA	ALC0790103NMA	CÓDIGO DA TURMA	3BCCNMA	NOTA	

ATENÇÃO:

- A avaliação somente poderá ser entregue depois de decorridos 50min de seu início.
 - Caneta esferográfica azul ou preta. Provas entregues escritas a lápis **NÃO** serão corrigidas.
 - Será atribuída nota zero ao aluno que devolver sua prova em branco, independentemente de ter assinado a Ata de Prova.
- Ao aluno flagrado **utilizando meios ilícitos ou não autorizados pelo professor para responder a avaliação** será atribuída nota. zero e, mediante representação do professor, responderá a Procedimento Administrativo Disciplinar, com base no Código de Ética.

O modelo abaixo foi construído para dar suporte a um sistema de Ouvidoria de uma empresa, onde o cliente faz o registro de uma ocorrência (reclamação, elogio, sugestão etc.). Ao fazer o registro, o ouvidor irá fazer a leitura e classificar a ocorrência por tipos, a fim de facilitar as análises posteriores, esse ouvidor ficará responsável por resolver a ocorrência. A cada ação desenvolvida para resolver a ocorrência, o ouvidor irá registrar como providência. Quando a ocorrência for resolvida, o ouvidor irá alterar a situação da ocorrência para “fechada”, pois toda ocorrência quando registrada terá a sua situação como “aberta”.



De acordo com a especificação acima, desenvolva:

- a) O script de geração do banco de dados, considere todas as restrições de integridade e consistência dos dados (constraints) (2 pts.).

1ª AVALIAÇÃO

```

1 CREATE SEQUENCE seqTipoOcorrencia;
2 CREATE SEQUENCE seqProvidencia;
3 CREATE SEQUENCE seqOcorrencia;
4
5 CREATE TABLE Ouvidor(
6     matricula int,
7     nome varchar(100),
8     CONSTRAINT pk_Ouvidor PRIMARY KEY (matricula)
9 );
10
11 CREATE TABLE Cliente(
12     cpf varchar,
13     nome varchar(100),
14     CONSTRAINT pk_Cliente PRIMARY KEY (cpf)
15 );
16
17 CREATE TABLE TipoOcorrencia(
18     idtipo int DEFAULT(nextval('seqTipoOcorrencia')),
19     descricao varchar(100),
20     CONSTRAINT pk_TipoOcorrencia PRIMARY KEY (idtipo)
21 );
22
23 CREATE TABLE Ocorrencia(
24     idocorrencia int DEFAULT(nextval('seqTipoOcorrencia')),
25     descocorrencia varchar(100),
26     situacao varchar(7) DEFAULT 'aberta',
27     dataocorrencia date,
28     matricula int,
29     cpf varchar,
30     idtipo int,
31     CONSTRAINT pk_Ocorrencia PRIMARY KEY (idocorrencia),
32     CONSTRAINT ck_situacao CHECK (situacao = 'aberta' OR situacao = 'fechada'),
33     CONSTRAINT fk_Ouvidor FOREIGN KEY (matricula) REFERENCES Ouvidor (matricula),
34     CONSTRAINT fk_Cliente FOREIGN KEY (cpf) REFERENCES Cliente (cpf),
35     CONSTRAINT fk_TipoOcorrencia FOREIGN KEY (idtipo) REFERENCES TipoOcorrencia (idtipo)
36 );
37
38 CREATE TABLE Providencia(
39     idprovidencia int DEFAULT(nextval('seqTipoOcorrencia')),
40     descprovidencia varchar(100),
41     idocorrencia int,
42     CONSTRAINT pk_Providencia PRIMARY KEY (idprovidencia),
43     CONSTRAINT fk_Ocorrencia FOREIGN KEY (idocorrencia) REFERENCES Ocorrencia (idocorrencia)
44 );
45

```

CREATE TABLE

Query returned successfully in 481 msec.

b) Escreva sentenças SQL para:

1. Projetar o nome do cliente, a data da ocorrência, a descrição da ocorrência e o nome do ouvidor, somente para as ocorrências que estão em aberto e foram registradas em 2020 (2 pts.).

```

-- 1
SELECT c.nome, oc.dataocorrencia, oc.descocorrencia, ov.nome
FROM Ocorrencia oc
    INNER JOIN Cliente c ON c.cpf = oc.cpf
    INNER JOIN Ouvidor ov On ov.matricula = oc.matricula
WHERE oc.situacao LIKE 'aberta'
    AND EXTRACT(YEAR from oc.dataocorrencia) = 2020;

```

	nome character varying (100)	dataocorrencia date	descocorrencia character varying (100)	nome character varying (100)
1	Isabela	2020-01-01	Mesa muito baixa	Diogo Felipe
2	Kevin	2020-04-02	Cadeiras ótimas	Ruan Gabriel
3	Dionisio	2020-05-17	Colocar fontes de vinho	Pedro Tavares
4	Perceus	2020-06-20	Falta um aquário	João Gabriel

1ª AVALIAÇÃO

2. Projetar o nome do ouvidor, a quantidade de ocorrências em aberto sob sua responsabilidade, a quantidade de ocorrências fechadas sob sua responsabilidade e o percentual dessas ocorrências fechadas em relação a todas as ocorrências sob a responsabilidade do ouvidor, somente para as ocorrências registradas em 2019 (2pts.).

```
SELECT ovp.nome,
(
    SELECT COUNT(oc.idocorrencia)
    FROM Ocorrencia oc
    INNER JOIN Ouvidor ovs ON ovs.matricula = oc.matricula
    WHERE ovs.matricula = ovp.matricula
    AND oc.situacao LIKE 'aberta'
    AND EXTRACT(YEAR from oc.dataocorrencia) = 2019
) ocorrencias_em_aberto,
(
    SELECT COUNT(oc.idocorrencia)
    FROM Ocorrencia oc
    INNER JOIN Ouvidor ovs ON ovs.matricula = oc.matricula
    WHERE ovs.matricula = ovp.matricula
    AND oc.situacao LIKE 'fechada'
    AND EXTRACT(YEAR from oc.dataocorrencia) = 2019
) ocorrencias_fechadas,
(
    round(1.0 * (
        SELECT COUNT(oc.idocorrencia)
        FROM Ocorrencia oc
        INNER JOIN Ouvidor ovs ON ovs.matricula = oc.matricula
        WHERE ovs.matricula = ovp.matricula
        AND oc.situacao LIKE 'fechada'
        AND EXTRACT(YEAR from oc.dataocorrencia) = 2019
    ) / (
        SELECT COUNT(oc.idocorrencia)
        FROM Ocorrencia oc
        INNER JOIN Ouvidor ovs ON ovs.matricula = oc.matricula
        WHERE ovs.matricula = ovp.matricula
        AND EXTRACT(YEAR from oc.dataocorrencia) = 2019
    ) * 100, 2)
) percentual
FROM Ouvidor ovp
```

	nome character varying (100)	ocorrencias_em_aberto bigint	ocorrencias_fechadas bigint	percentual numeric
1	Diogo Felipe	1	1	50.00
2	Ruan Gabriel	1	1	50.00
3	Pedro Tavares	1	1	50.00
4	João Gabriel	1	0	0.00

3. Projetar um ranking de tipos de ocorrência, com a descrição da ocorrência e a sua quantidade, ordenada de forma descendente da quantidade de ocorrências (2 pts.).

1ª AVALIAÇÃO

```
SELECT tocp.descricao,
(
    SELECT COUNT(oc.idtipo)
    FROM Ocorrencia oc
    INNER JOIN TipoOcorrencia tocs ON tocs.idtipo = oc.idtipo
    WHERE tocs.descricao = tocp.descricao
) quantidade
FROM TipoOcorrencia tocp
ORDER BY quantidade DESC;
```

	descricao	quantidade
	character varying (100)	bigint
1	reclamação	5
2	elogio	3
3	sugestão	3

4. Criar uma View que irá, quando chamada, retornar o cpf e o nome dos clientes que já fizeram algum tipo de ocorrência na ouvidoria. Tenha cuidado para que a View não retorne nomes repetidos (2 pts.).

```
CREATE VIEW v_cliente AS
(
    SELECT DISTINCT c.cpf, c.nome
    FROM Cliente c
    INNER JOIN Ocorrencia oc ON c.cpf = oc.cpf
)
```

CREATE VIEW

Query returned successfully in 222 msec.

5. Qual o Ouvidor campeão na resolução de problemas? Faça uma sentença para nos mostrar o seu nome (2 pts.).

```
SELECT ovp.nome,
(
    SELECT COUNT(oc.idocorrencia)
    FROM Ocorrencia oc
    INNER JOIN Ouvidor ovs ON ovs.matricula = oc.matricula
    WHERE ovs.matricula = ovp.matricula
    AND oc.situacao LIKE 'fechada'
) ocorrencias_fechadas
FROM Ouvidor ovp
ORDER BY ocorrencias_fechadas DESC
LIMIT 1;
```

	nome	ocorrencias_fechadas
	character varying (100)	bigint
1	Diogo Felipe	1

OBS:

1ª AVALIAÇÃO

- 1) Você pode consultar o material gerado durante as aulas.
- 2) Todos os códigos e seus RESULTADOS devem ser copiados neste documento WORD e postados no Teams. Caso tenha dificuldades de postar, envie para o email:
almir.monteiro.junior@gmail.com

BOA PROVA