

	<b>Universidade Federal Fluminense</b>
	<b>Disciplina:</b> Banco de Dados - EAD05014
	<b>Professor:</b> Daniel de Oliveira e Marcos Bedo
	<b>Semestre:</b> 2024.1
	<b>Curso:</b> Tecnologia em Sistemas de Computação

## AP2

### Leia as Instruções:

- A prova está feita para ser executada em 2 horas, a mão.
- A prova é **INDIVIDUAL** e **SEM CONSULTA**.
- É essencial ter comportamento **ÉTICO**. Não copie respostas de colegas!
- Não esqueça de preencher todas as respostas no cartão de respostas.

Considere o esquema relacional a seguir de uma empresa de aluguel de automóveis simplificada para responder às questões de 1 a 7. As chaves primárias se encontram sublinhadas. A relação **RESERVA** apresenta as reservas de automóveis realizadas por um(a) determinado(a) cliente. Em uma mesma reserva, o(a) cliente pode reservar diversos automóveis.

```

CLIENTE(cpf, nome, municipio, telefone) % CPF, nome, município e telefone do(a) cliente
AUTOMOVEL(chassi, placa, modelo, marca, ano_fabricacao, ano_modelo) % Chassi, placa, modelo, marca, ano de fabricação e do modelo do automóvel
RESERVA(rid, cpf, datainicio, datatermino, multa) % Identificador da reserva, cpf do(a) cliente, data de início e de fim do aluguel e o preço da multa em caso de atraso
      cpf referencia CLIENTE(cpf)
RESERVA AUTOMOVEL(rid,chassi)
      rid referencia RESERVA(rid)
      chassi referencia AUTOMOVEL(chassi)

```

1. Apresente o comando SQL que retorne o nome dos clientes que nunca reservaram um automóvel. **(1,0 pontos)**

```
SELECT c.nome FROM CLIENTE c where c.cpf NOT IN (SELECT r.cpf FROM RESERVA r);
```

Variações usando **LEFT JOIN** também são válidas.

2. Apresente o comando SQL que retorne o modelo do automóvel, sua placa, as datas de início e término e o preço da multa paga (caso seja maior que zero) de cada uma das reservas realizadas para cada automóvel. **(1,0 pontos)**

```
SELECT a.modelo, a.placa, r.datainicio, r.datatermino, r.multa  
FROM RESERVA r, RESERVA_AUTOMOVEI ra, AUTOMOVEI a WHERE r.rid  
= ra.rid AND ra.chassi = a.chassi AND r.multa > 0;
```

Variações dessa consulta usando *sub-queries* também foram aceitas.

3. Apresente o comando SQL que exclui a tabela **RESERVA**. Observação: o objetivo é excluir a tabela, não somente suas tuplas. **(1,0 pontos)**

```
DROP TABLE RESERVA;
```

4. Apresente um comando SQL que insere a tupla ('9BD111060T5002156', 'RTP09B6', 'Ford Focus', 2023, 2024) na relação **AUTOMOVEI**. Observação: Repare que o valor do atributo 'marca' não foi informado. **(1,0 pontos)**

```
INSERT INTO AUTOMOVEI(chassi, placa, modelo, marca, ano_  
fabricacao, ano_modelo) VALUES ('9BD111060T5002156', 'RTP09B6',  
'Ford Focus', 2023, 2024);
```

5. Considere que a tabela **CLIENTE** possui o campo **nome**. Apresente o comando SQL que lista os nomes dos clientes que possuem os sobrenomes **SILVA** em conjunto com **SOUZA** ou **SOUSA**, sendo que **SILVA** deve ser o primeiro sobrenome. **(1,0 pontos)**

ANULADA. Os pontos foram distribuídos nas demais questões.

6. Apresente um comando em SQL que exclua os clientes que não possuam reservas de automóveis da marca 'FIAT' ou 'CITROEN'. Assuma que a tabela **CLIENTE** foi criada com a opção **ON DELETE CASCADE**. **(1,0 pontos)**

```
DELETE FROM CLIENTE c WHERE c.cpf NOT IN (SELECT c1.cpf FROM  
CLIENTE c1, RESERVA r, RESERVA_AUTOMOVEI ra, AUTOMOVEI a  
WHERE r.cpf = c1.cpf AND r.rid = ra.rid  
AND ra.chassi = a.chassi AND (a.marca = 'FIAT' OR a.marca  
= 'CITROEN'));
```

7. Apresente um comando em SQL que atualize as tuplas da tabela **RESERVA** dando uma redução de 20% no valor de todas as multas maiores que R\$500,00 de clientes do município de Niterói e São Gonçalo. (1,0 pontos)

```
UPDATE RESERVA r SET r.multa = r.multa *0.8 WHERE r.multa > 500 AND r.cpf IN (SELECT c.cpf FROM CLIENTE c WHERE UPPER(c.municipio) = 'NITEROI' OR UPPER(c.municipio) = 'SÃO GONÇALO' );
```

8. Considere a relação **R(ID, Nome, UF, Salario)**. A chave primária se encontra sublinhada. Elabore uma consulta em SQL que apresente a média dos salários por estado (UF), mas somente quando a média for superior a R\$ 5.000,00 e para pessoas que possuem **OLIVEIRA** no sobrenome (considere que o campo **NOME** é um VARCHAR que engloba tanto o nome quanto o sobrenome da pessoa. Além disso, o resultado deve estar ordenado por UF de forma alfabética. (1,5 pontos)

```
SELECT UF, AVG(Salario)
FROM R
WHERE UPPER(Nome) LIKE '%OLIVEIRA%'
GROUP BY UF
HAVING AVG(Salario) > 5000
ORDER BY UF;
```

9. Analise as relações apresentadas a seguir e informe em qual forma normal elas se encontram (1FN, 2FN ou 3FN). Justifique suas respostas. As chaves primárias de cada relação se encontram sublinhadas. Assuma que todos os professores em um mesmo nível da carreira ganham o mesmo salário. (1,5 pontos)

- (a) **PROFESSOR(CodDep<sup>to</sup>, NomeDep<sup>to</sup>, CodProfessor, NomeProfessor)**

Se encontra na 1FN, pois não tem relações aninhadas, mas ainda existe dependência parcial (*i.e.*, NomeProfessor depende **apenas** de CodProfessor).

- (b) **PROFESSOR(CodProfessor, NomeProfessor, Salario)**

Se encontra na 3FN, pois não há dependências multivaloradas, nem dependências parciais e nem dependência transitivas.

- (c) **DEPARTAMENTO(CodDep<sup>to</sup>, NomeDep<sup>to</sup>, (CodProfessor, NomeProfessor)**

Não se encontra na 1FN, pois há relações aninhadas (*i.e.*, existe uma dependência multivalorada  $\text{CodDep}^{\text{to}} \rightarrow \{\text{CodProfessor}, \text{NomeProfessor}\}$ ).

- (d) **PROFESSOR(CodProfessor, NomeProfessor, NivelCarreira, Salario)**

Se encontra na 2FN, pois há dependência transitiva, uma vez que o salário depende funcionalmente de NivelCarreira.