

Aluno

Nº Matrícula

Modalidade:
PRESENCIAL

Turno:
NOITE

Disciplina:
BANCO DE DADOS II

Prova nº

Professor:
Dr. Argemiro Pentian Junior

Data
09/10/2025

Questão 1: Stored Procedures (2 pontos)

Crie uma stored procedure chamada `ListarAlunosCurso` que recebe o `id_curso` (IN) e retorna uma lista com os nomes dos alunos matriculados nesse curso.

- ✓ Use a tabela `Alunos` e exiba apenas o campo `nome_aluno`.
- ✓ Inclua tratamento para caso o curso não exista (retorne uma mensagem usando SELECT).
- ✓ Escreva o código completo, incluindo DELIMITER.

Questão 2: Views (2 pontos)

Crie uma view chamada `ListaProfessoresDepartamentos` that exiba o nome do professor e o nome do departamento ao qual ele está associado.

- ✓ Use JOIN entre as tabelas `Professores` e `Departamentos`.
- ✓ Explique brevemente por que essa view seria útil (ex.: para relatórios administrativos).
- ✓ Escreva o código SQL completo para criar a view.

Questão 3: Triggers (Intermediário - 2 pontos)

Crie um trigger chamado `ValidarNotaAntesInserir` do tipo BEFORE INSERT na tabela `Notas`.

- ✓ O trigger deve verificar se o `valor_nota` é maior que 10.0 ou menor que 0.0, lançando um erro com SIGNAL caso seja inválido.
- ✓ Se válido, permita a inserção sem alterações.
- ✓ Escreva o código completo, incluindo DELIMITER.

Questão 4: Correção de Erro - Transações (2 pontos)

O código abaixo tenta registrar uma nova matrícula na tabela `Matriculas`, mas contém erros que podem causar inconsistências. Identifique pelo menos 2 erros, explique cada um e forneça o código SQL corrigido.

-- Código com erro:

```
BEGIN TRANSACTION
INSERT INTO Matriculas (id_aluno id_turma, data_matricula) VALUES (1, 1, '2025-01-01');
COMMIT;
```

Questão 5: Correção de Erro - Stored Procedures e Triggers (2 pontos)

O código abaixo tenta criar uma stored procedure para inserir uma nota e um trigger para logar a inserção, mas contém erros. Identifique pelo menos 3 erros, explique cada um e forneça o código SQL corrigido completo (procedure e trigger).

-- Código com erro:

```
DELIMITER //
CREATE PROCEDURE InserirNota (
    IN p_id_matricula INT
    IN p_valor_nota REAL,
    OUT p_mensagem VARCHAR(255)
)
```

```
BEGIN
    IF p_valor_nota > 10 THEN
        SET p_mensagem = 'Nota inválida!';
        SIGNAL SQLSTATE '45000' MESSAGE_TEXT = p_mensagem;
    ELSE
        INSERT INTO Notas (id_matricula valor_nota) VALUES (p_id_matricula, p_valor_nota);
        SET p_mensagem = 'Nota inserida!';
    END IF
END //
DELIMITER ;

-- Trigger implícito (errado)
CREATE TRIGGER LogNota AFTER INSERT ON Notas
BEGIN
    INSERT INTO LogNotas (mensagem) VALUES ('Nova nota inserida');
END;
```