

SQL DDL, DML e DQL

Considerações:

- Esta avaliação **vale 2,5 pontos** para a avaliação continuada A2.
- O arquivo enviado no Blackboard deve ser um **arquivo de texto puro com extensão .sql**; não serão aceitos arquivos com formatação (.docx, .rtf, .odt etc.)
- Os comandos devem ser apresentados com indentação adequada.
- Os comandos em resposta a cada questão devem estar indicados no *script* com comentário. Ex.:


```
-- Comandos da questão 1:  
CREATE DATABASE HotelSaintGermain;  
USE HotelSaintGermain;
```

Contextualizando a avaliação (esta avaliação é continuação do estudo de caso da avaliação A2.1):

- De acordo com a avaliação anterior, sua equipe foi designada para projetar uma base de dados para um sistema (estudo de caso) de um domínio (tema) qualquer.
- Verifique a documentação de entrega para garantir que seu projeto lógico tenha atendido às especificações para prosseguir com esta entrega.

Para o estudo de caso apresentado, **elabore scripts SQL para:**

- 1) Criar a base de dados e prepará-la para uso. (0,2 pt)
- 2) Criar as tabelas de acordo com o modelo lógico relacional idealizado na avaliação anterior, considerando as restrições de integridade. (0,6 pt)
- 3) Inserir pelo menos 5 registros em cada tabela (INSERT). (0,2 pt)
- 4) Elaborar pelo menos 10 necessidades negociais que a base deve atender, considerando que: (1,5 pt)
 - ✓ cada tabela da base tenha sido envolvida em pelo menos uma necessidade;
 - ✓ os comandos para atender às necessidades devem considerar as outras 3 operações básicas (SELECT, UPDATE e DELETE);
 - ✓ pelo menos dois dos comandos deve utilizar JOIN;
 - ✓ pelo menos um dos comandos deve utilizar GROUP BY com alguma fórmula de agregação;
 - ✓ para cada necessidade, descrição da solicitação deve estar indicada no *script* como linha de comentário.

Seguem 3 exemplos de necessidades, adequadamente indicadas no *script*, e possível resposta para cada uma delas (os exemplos consideram o caso do Hotel Saint Germain):

-- Comandos da questão 4:

-- Necessidade 1: Listar, em ordem alfabética (crescente) de nome, o nome e a data de nascimento de todos os clientes cujo nome inicia com a letra 'A'.

```
SELECT Nome, DtaNasc FROM Cliente WHERE Nome LIKE 'A%' ORDER BY Nome;
```

-- Necessidade 2: Aumentar em 10% o valor da diária de todos os quartos acima do segundo andar (inclusive).

```
UPDATE Quarto SET VlrDiaria = VlrDiaria * 1.1 WHERE Andar >= 2;
```

-- Necessidade 3: Listar o número do quarto e a quantidade de vezes que ele foi ocupado desde o ano de 2015 (inclusive); listar o quarto mesmo que ele nunca tenha sido ocupado.

```
SELECT Q.Numero AS Quarto, Count(O.ReservaNumero) AS Vezes  
FROM Quarto Q LEFT JOIN Reserva R ON R.QuartoNumero = Q.Numero  
LEFT JOIN Ocupacao O ON O.ReservaNumero = R.Numero  
WHERE Year(O.Entrada) >= 2015  
GROUP BY Quarto ORDER BY Quarto;
```