

Instruções gerais:

1. Utilize o SQL Server Management Studio (SSMS) para escrever os Scripts SQL.
2. Desenvolva a solução de cada exercício em um arquivo diferente (exe01.sql, exe02.sql, ...)
3. Ao concluir os exercícios, envie todos os códigos criados para o Repositório criado na sua conta pessoal do Github, em uma pasta identificada com o tema da sessão
4. Envie o link do repositório no Github ou a Captura de tela (print) do repositório, conforme a orientação do instrutor através da Plataforma Canvas na data indicada.

Mantenha a entrega das Atividades em dia na Plataforma Canvas

EXERCÍCIOS

Boas práticas:

1. Resolva **03 exercícios da lista**.
2. Leia o enunciado do exercício com atenção
3. Observe as indicações de Entrada e Saída esperadas em cada exercício
4. Observe com atenção os desenhos e diagramas inseridos nos exercícios para facilitar a compreensão
5. Utilize o Cookbook, os Vídeos da Plataforma e os Códigos guia como referências para a resolução dos exercícios
6. Caso ainda fique alguma dúvida, consulte os instrutores da sua turma pelo Discord

ATIVIDADE 01

Crie um banco de dados para um serviço de um Games Online. O nome do Banco de dados deverá ter o seguinte nome **db_generation_game_online**. O sistema trabalhará com as informações dos personagens do jogo. O sistema trabalhará com 2 tabelas **tb_personagens** e **tb_classes**, que deverão estar relacionadas.



Boas Práticas:

1. Crie a tabela **tb_classes** e determine pelo menos 2 atributos, além da **Chave Primária**, relevantes para classificar os personagens do Game Online.
2. Crie a tabela **tb_personagens** e determine 4 atributos, além da **Chave Primária**, relevantes aos personagens do Game Online.
3. **Não esqueça de criar a Foreign Key da tabela tb_classes na tabela tb_personagens.**
4. Insira 5 registros na tabela **tb_classes**.
5. Insira 8 registros na tabela **tb_personagens**, preenchendo a Chave Estrangeira para criar a relação com a tabela **tb_classes**.
6. Faça um **SELECT** que retorne todos os personagens cujo poder de ataque seja maior do que 2000.
7. Faça um **SELECT** que retorne todos os personagens cujo poder de defesa esteja no intervalo 1000 e 2000.
8. Faça um **SELECT** utilizando o operador **LIKE**, buscando todos os personagens que possuam a letra C no atributo nome.
9. Faça um **SELECT** utilizando a cláusula **INNER JOIN**, unindo os dados da tabela **tb_personagens** com os dados da tabela **tb_classes**.

10. Faça um SELECT utilizando a cláusula INNER JOIN, unindo os dados da tabela tb_personagens com os dados da tabela tb_classes, onde traga apenas os personagens que pertençam a uma classe específica (Exemplo: todos os personagens da classe dos arqueiros).
11. Salve todas as queries para cada um dos requisitos do exercício em um único script (arquivo .SQL) e coloque no seu Github pessoal, no repositório que você criou sobre Banco de dados.

Atividade 2

Crie um banco de dados para um serviço de uma Pizzaria. O nome do Banco de dados deverá ter o seguinte nome **db_pizzaria_legal**. O sistema trabalhará com as informações dos produtos comercializados pela empresa. O sistema trabalhará com 2 tabelas **tb_pizzas** e **tb_categorias**, que deverão estar relacionadas.



Boas Práticas:

1. Crie a tabela **tb_categorias** e determine pelo menos 2 atributos, além da **Chave Primária**, relevantes para classificar as pizzas.
2. Crie a tabela **tb_pizzas** e determine 4 atributos, além da **Chave Primária**, relevantes aos produtos da pizzaria.
3. **Não esqueça de criar a Foreign Key da tabela tb_categorias na tabela tb_pizzas.**
4. Insira 5 registros na tabela **tb_categorias**.
5. Insira 8 registros na tabela **tb_pizzas**, preenchendo a Chave Estrangeira para criar a relação com a tabela **tb_categorias**.
6. Faça um **SELECT** que retorne todas as pizzas cujo valor seja maior do que R\$ 45,00.
7. Faça um **SELECT** que retorne todas as pizzas cujo valor esteja no intervalo R\$ 50,00 e R\$ 100,00.
8. Faça um **SELECT** utilizando o operador **LIKE**, buscando todas as pizzas que possuam a letra M no atributo nome.
9. Faça um **SELECT** utilizando a cláusula **INNER JOIN**, unindo os dados da tabela **tb_pizzas** com os dados da tabela **tb_categorias**.

10. Faça um SELECT utilizando a cláusula INNER JOIN, unindo os dados da tabela tb_pizzas com os dados da tabela tb_categorias, onde traga apenas as pizzas que pertençam a uma categoria específica (Exemplo: Todas as pizzas que são doce).
11. Salve todas as queries para cada um dos requisitos do exercício em um único script (arquivo .SQL) e coloque no seu Github pessoal, no repositório que você criou sobre Banco de dados.

Atividade 3

Crie um banco de dados para um serviço de uma Farmácia. O nome do Banco de dados deverá ter o seguinte nome **db_farmacia_bem_estar**. O sistema trabalhará com as informações dos produtos comercializados pela empresa. O sistema trabalhará com 2 tabelas **tb_produtos** e **tb_categorias**, que deverão estar relacionadas.



Boas Práticas:

1. Crie a tabela **tb_categorias** e determine pelo menos 2 atributos, além da **Chave Primária**, relevantes para classificar os produtos.
2. Crie a tabela **tb_produtos** e determine 4 atributos, além da **Chave Primária**, relevantes aos produtos da farmácia.
3. **Não esqueça de criar a Foreign Key da tabela tb_categorias na tabela tb_produtos.**
4. Insira 5 registros na tabela tb_categorias.
5. Insira 8 registros na tabela tb_produtos, preenchendo a Chave Estrangeira para criar a relação com a tabela tb_categorias.
6. Faça um SELECT que retorne todos os produtos cujo valor seja maior do que R\$ 50,00.
7. Faça um SELECT que retorne todos os produtos cujo valor esteja no intervalo R\$ 5,00 e R\$ 60,00.
8. Faça um SELECT utilizando o operador LIKE, buscando todos os produtos que possuam a letra C no atributo nome.
9. Faça um SELECT utilizando a cláusula INNER JOIN, unindo os dados da tabela tb_produtos com os dados da tabela tb_categorias.
10. Faça um SELECT utilizando a cláusula INNER JOIN, unindo os dados da tabela tb_produtos com os dados da tabela tb_categorias, onde traga apenas os produtos

que pertençam a uma categoria específica (Exemplo: todos os produtos que pertencem à categoria Higiene Pessoal).

11. Salve todas as queries para cada um dos requisitos do exercício em um único script (arquivo .SQL) e coloque no seu Github pessoal, no repositório que você criou sobre Banco de dados.

Atividade 4

Crie um banco de dados para um serviço de um Açougue ou de um Hortifruti. O nome do Banco de dados deverá ter o seguinte nome **db_cidade_das_carnes** ou **cidade_dos_vegetais**. O sistema trabalhará com as informações dos produtos comercializados pela empresa. O sistema trabalhará com 2 tabelas **tb_produtos** e **tb_categorias**, que deverão estar relacionadas.



Boas Práticas:

1. Crie a tabela **tb_categorias** e determine pelo menos 2 atributos, além da **Chave Primária**, relevantes para classificar os produtos.
2. Crie a tabela **tb_produtos** e determine 4 atributos, além da **Chave Primária**, relevantes aos produtos da farmácia.
3. **Não esqueça de criar a Foreign Key da tabela tb_categorias na tabela tb_produtos.**
4. Insira 5 registros na tabela **tb_categorias**.
5. Insira 8 registros na tabela **tb_produtos**, preenchendo a Chave Estrangeira para criar a relação com a tabela **tb_categorias**.
6. Faça um **SELECT** que retorne todos os produtos cujo valor seja maior do que R\$ 50,00.
7. Faça um **SELECT** que retorne todos os produtos cujo valor esteja no intervalo R\$ 50,00 e R\$ 150,00.
8. Faça um **SELECT** utilizando o operador **LIKE**, buscando todos os produtos que possuam a letra C no atributo nome.
9. Faça um **SELECT** utilizando a cláusula **INNER JOIN**, unindo os dados da tabela **tb_produtos** com os dados da tabela **tb_categorias**.

10. Faça um SELECT utilizando a cláusula INNER JOIN, unindo os dados da tabela tb_produtos com os dados da tabela tb_categorias, onde traga apenas os produtos que pertençam a uma categoria específica (Exemplo: todos os produtos que pertencem à categoria aves ou frutas).
11. Salve todas as queries para cada um dos requisitos do exercício em um único script (arquivo .SQL) e coloque no seu Github pessoal, no repositório que você criou sobre Banco de dados.

Atividade 5

Crie um banco de dados para um serviço de uma loja de Material de Construção. O nome do Banco de dados deverá ter o seguinte nome **db_construindo_vidas**. O sistema trabalhará com as informações dos produtos comercializados pela empresa. O sistema trabalhará com 2 tabelas **tb_produtos** e **tb_categorias**, que deverão estar relacionadas.



Boas Práticas:

1. Crie a tabela **tb_categorias** e determine pelo menos 2 atributos, além da **Chave Primária**, relevantes para classificar os produtos.
2. Crie a tabela **tb_produtos** e determine 4 atributos, além da **Chave Primária**, relevantes aos produtos da farmácia.
3. **Não esqueça de criar a Foreign Key da tabela tb_categorias na tabela tb_produtos.**
4. Insira 5 registros na tabela tb_categorias.
5. Insira 8 registros na tabela tb_produtos, preenchendo a Chave Estrangeira para criar a relação com a tabela tb_categorias.
6. Faça um SELECT que retorne todos os produtos cujo valor seja maior do que R\$ 100,00.
7. Faça um SELECT que retorne todos os produtos cujo valor esteja no intervalo R\$ 70,00 e R\$ 150,00.
8. Faça um SELECT utilizando o operador LIKE, buscando todos os produtos que possuam a letra C no atributo nome.
9. Faça um SELECT utilizando a cláusula INNER JOIN, unindo os dados da tabela tb_produtos com os dados da tabela tb_categorias.
10. Faça um SELECT utilizando a cláusula INNER JOIN, unindo os dados da tabela tb_produtos com os dados da tabela tb_categorias, onde traga apenas os produtos

que pertençam a uma categoria específica (Exemplo: todos os produtos que pertencem à categoria hidráulica).

11. Salve todas as queries para cada um dos requisitos do exercício em um único script (arquivo .SQL) e coloque no seu Github pessoal, no repositório que você criou sobre Banco de dados.

Atividade 6

Crie um banco de dados para uma plataforma de Cursos Online (EAD). O nome do Banco de dados deverá ter o seguinte nome **db_curso_da_minha_vida**. O sistema trabalhará com as informações dos produtos comercializados pela empresa. O sistema trabalhará com 2 tabelas **tb_cursos** e **tb_categorias**, que deverão estar relacionadas.



Boas Práticas:

1. Crie a tabela **tb_categorias** e determine pelo menos 2 atributos, além da **Chave Primária**, relevantes para classificar os produtos.
2. Crie a tabela **tb_cursos** e determine 4 atributos, além da **Chave Primária**, relevantes aos produtos da farmácia.
3. **Não esqueça de criar a Foreign Key da tabela tb_categorias na tabela tb_cursos.**
4. Insira 5 registros na tabela **tb_categorias**.
5. Insira 8 registros na tabela **tb_cursos**, preenchendo a Chave Estrangeira para criar a relação com a tabela **tb_categorias**.
6. Faça um **SELECT** que retorne todos os cursos cujo valor seja maior do que R\$ 500,00.
7. Faça um **SELECT** que retorne todos os cursos cujo valor esteja no intervalo R\$ 600,00 e R\$ 1000,00.
8. Faça um **SELECT** utilizando o operador **LIKE**, buscando todos os cursos que possuam a letra J no atributo nome.
9. Faça um **SELECT** utilizando a cláusula **INNER JOIN**, unindo os dados da tabela **tb_cursos** com os dados da tabela **tb_categorias**.
10. Faça um **SELECT** utilizando a cláusula **INNER JOIN**, unindo os dados da tabela **tb_cursos** com os dados da tabela **tb_categorias**, onde traga apenas os produtos

que pertençam a uma categoria específica (Exemplo: todos os cursos que pertencem à categoria Java).

11. Salve todas as queries para cada um dos requisitos do exercício em um único script (arquivo .SQL) e coloque no seu Github pessoal, no repositório que você criou sobre Banco de dados.