

## ATIVIDADE – MySQL

Atividade prática - Banco de dados Relacional

### Instruções gerais:

1. Utilize o MySQL Workbench para escrever os Scripts SQL.
2. Desenvolva a solução de cada exercício em um arquivo (exe01.sql, exe02.sql, ...)
3. Ao concluir os scripts SQL, envie todos para o Repositório sobre Banco de dados criado na sua conta pessoal do Github
4. Envie o link do repositório no Github através da Plataforma da Generation na data indicada
5. Caso seja solicitado, adicione os links individuais dos scripts SQL indicados, no item:  
**Adicione um dos links da sua entrega**, localizada depois do link do Repositório, na tela de entrega da atividade na plataforma, para validação da atividade.

**Mantenha as entregas das Atividades em dia na Plataforma da Generation**

## ATIVIDADES PRÁTICAS

### 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 a categoria cosméticos).

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 Hortifrut. 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 a 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 a 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 a 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.