

Criação de relacionamentos em SQL – Cláusula CONSTRAINT (restrição) 1

- A linguagem SQL permite o uso da cláusula CONSTRAINT para a especificação de um campo que é chave estrangeira em uma determinada tabela;
- O uso de restrição reforça a segurança em operações de exclusão e alteração de registros em tabelas relacionadas – integridade referencial;
- As tabelas com chave estrangeira devem sempre ser **criadas DEPOIS** da tabela original, aquela que “emprestou” uma coluna para funcionar como chave estrangeira na tabela dependente. Exemplo: se temos uma tabela médico e uma tabela paciente, onde o id do médico é chave estrangeira na tabela paciente, a tabela médico deve ser criada ANTES de paciente;
- Da mesma forma, uma tabela com chave estrangeira deve sempre ser **apagada ANTES** da tabela principal.

```
➔ CREATE TABLE IF NOT EXISTS nome_da_tabela(  
    campo1 tipo_de_dado [definições do campo],  
    campo2 tipo_de_dado [definições do campo],  
    campo_chave_estrangeira tipo_de_dado [definições do campo],  
  
    FOREIGN KEY (campo_chave_estrangeira) REFERENCES  
    nome_da_tabela_pai (campo_da_tabela_pai)  
);
```

CONSTRAINT - exemplo

➤ A tabela médico contém o campo id, que é sua chave primária, e que será chave estrangeira da tabela paciente;

➤ Médico deve ser criada antes de paciente;

➤ CREATE TABLE IF NOT EXISTS **medico**(
 Id INT PRIMARY KEY,
 Nome VARCHAR(100));

➤ CREATE TABLE IF NOT EXISTS paciente(
 Nome VARCHAR(100),
 Id_medico INT,
 FOREIGN KEY (Id_medico) REFERENCES medico(id));