

Laboratório SQL - DDL - Gerando o script SQL de criação do banco

Giulia Moura Ferreira, 200018795

Ambos os scripts podem ser encontrados através deste [\[link\]](#)

1. Estrutura Geral

O **script 1 “MeuScript.sql”** apresenta uma estrutura melhor planejada, definindo claramente os atributos das tabelas, com tipos de dados apropriados e relacionamentos diretos entre as entidades. Já o **script 2 “ScriptBRModelo.sql”** demonstra uma abordagem inconsistente, com nomes inadequados (o que acabou sendo culpa minha e não do aplicativo, pois eu defini os nomes com espaços no diagrama, o que causaria erro na execução do script), falta de padronização e problemas de definição que comprometem sua funcionalidade.

2. Tipos de Dados

No script 1, os tipos de dados foram escolhidos de acordo com a conveniência dos atributos, por exemplo:

- Atributos de texto, como Nome e Endereço, usam *varchar*.
- Datas utilizam o tipo *date*, que é ideal para operações de manipulação de datas.
- Atributos categóricos, como sexo, utilizam *char(1)* (F ou M).

Enquanto no script 2, todos os atributos foram definidos como *int*, o que é completamente inadequado para dados como nomes, datas ou descrições.

3. Chaves Primárias (PKs)

O script 1 define as chaves primárias de forma lógica para cada tabela, em Infração, por exemplo, utiliza uma PK composta (*data_hora*, *idVeiculo*, *idTipo_Infracao*), o que garante a unicidade dos registros ao considerar o veículo, o momento e o tipo de infração.

Já o script 2 apresenta erros graves na definição da chave primária composta, pois ele define várias PKs em vez de uma única PK composta. Isso viola as regras de integridade.

4. Chaves Estrangeiras (FKs)

No script 1, as FKs são declaradas diretamente nas tabelas, garantindo relacionamentos claros e consistentes entre as entidades.

No script 2, as FKs são definidas posteriormente usando *ALTER TABLE*. Embora essa abordagem seja tecnicamente válida, no caso específico do script, ela foi mal executada, porque as FKs estão apontando para tabelas ou colunas incorretas, como relacionar *idVeiculo* a *Proprietário*, em vez de *Veículo*. Isso compromete a integridade referencial do banco e torna o script inviável para execução.

5. Relacionamentos

O script 1 representa corretamente os relacionamentos entre as tabelas, como:

- **Proprietário e Veículo:** Um proprietário pode possuir vários veículos.
- **Veículo e Categoria/Modelo:** Cada veículo está associado a uma única categoria e modelo.
- **Infração e outras tabelas (local, agente, etc.):** Relacionamentos claros e funcionais.

O script 2 apresenta relacionamentos confusos e errados:

- FK de **Infração** aponta para tabelas e colunas equivocadas.
- Não há clareza sobre os relacionamentos entre entidades, dificultando o entendimento do modelo.

6. Sintaxe

O script 1 segue uma sintaxe padronizada, incluindo a terminação de comandos com “;” e comentários úteis, como:

```
Sexo char(1),    -- F ou M
```

Enquanto o script 2 apresenta erros de sintaxe que causariam erro na execução:

- Falta de “;” após comandos *ALTER TABLE*.
- Presença de vírgulas no final das definições de tabelas, como:

```
Velocidade permitida INT,  
);
```