

UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

João Lucas Pereira e Sousa - 10994311

Joel Felipe Coelho - 4865826

Ryan Souza Sá teles - 12822062

Igor Mateus Queiroz Gato - 10261481

**TRABALHO PRÁTICO T1 - FEDERAÇÃO INTERNACIONAL
DE AUTOMOBILISMO**

Exercícios de DDL e DML

SCC0541 - LABORATÓRIO DE BASES DE DADOS
Prof. Caetano Traina Júnior

São Carlos
2023

1. DESCRIÇÃO DO PROBLEMA E DOS REQUISITOS DE DADOS

Neste trabalho, deverão ser feitas duas atividades:

- Elaboração do script em SQL para criação das tabelas no banco de dados.
- Elaboração de script SQL para carga dos dados no banco de dados.

Ambas devem ser executadas no SGBD escolhido.

Para a construção das tabelas, deverá ser utilizado o mapeamento disponibilizado no Tidia no arquivo Mapeamento-atividade.PDF. Para a carga dos dados, deverão ser usados os arquivos mencionados anteriormente e disponibilizados no Google Drive.

2. MAPEAMENTO DA UNIDADE RELACIONAL PARA CRIAÇÃO DO DDL E DML

Ao se considerar transformar um mapeamento relacional em um DML e DDL, é importante levar em consideração alguns pontos principais que necessitam estar contidos para que seja realizado com sucesso.

Uma chave primária é necessária para todas as tabelas (ou relações): Um campo ou conjunto de campos conhecido como chave primária identifica exclusivamente cada registro em uma tabela. É usado para garantir a integridade dos dados e permitir relacionamentos entre tabelas.

As tabelas precisam ser normalizadas para evitar dados redundantes: A organização de dados em um banco de dados é feita por meio da normalização. Ela envolve a criação de tabelas e a conexão dessas tabelas de acordo com regras destinadas a proteger os dados e aumentar a flexibilidade do banco de dados, eliminando informações redundantes e dependências inconsistentes.

Para manter a integridade referencial dos dados, as chaves estrangeiras devem ser definidas: Em um banco de dados relacional, os relacionamentos entre tabelas são criados usando chaves estrangeiras. Elas são usadas para verificar o relacionamento entre os dados em uma tabela e os dados em outra tabela.

Para garantir precisão e eficiência no armazenamento de dados, é preciso escolher cuidadosamente os tipos de dados: Os tipos de dados são usados para definir o tipo de informação que pode ser armazenada em um campo da tabela. É importante escolher o tipo correto de dados para garantir que os dados sejam armazenados com precisão e eficiência.

Para garantir que os dados armazenados sejam válidos e consistentes, as restrições de integridade devem ser definidas: As restrições de integridade são usadas para garantir que os dados armazenados em uma tabela sejam válidos e consistentes. Elas podem ser usadas para restringir o tipo de informação que pode ser inserida em um campo ou para garantir que os valores inseridos atendam a determinados critérios.

Limites de caracteres

Observando os dados presentes no banco de dados passado como parâmetro, limitar o número de caracteres que podem ser inseridos em entradas textuais pareceu a melhor opção pois nenhuma das inserções possui um conteúdo que exceda esse limite. Além disso, para futuras inserções, evita que os usuários insiram informações excessivamente longas ou inválidas.

Tipo de dados

O tipo de dado escolhido para ser utilizado em cada coluna e inserção levou em consideração o arquivo .CSV com os tipos de dados de cada coluna ali presente.

Atributos Não Nulos

Julgamos importante manter a obrigatoriedade de atributos que são utilizados pelas tabelas como chave primária e chave estrangeira impedindo que qualquer dado seja inserido nas tabelas sem que as mesmas estejam presentes.

A maioria das entidades são identificadas unicamente pelo seu id de inserção, chave primária, número incrementado automaticamente à medida que os dados são inseridos na tabela, com exceção da entidade SEASONS que utiliza year como sua identificação, LAPTIMES que utiliza além de seu id, id de driver e lap, e a entidade PITSTOP que utiliza id de RACE, DRIVER, além de stop. Já para as chaves estrangeiras, o mesmo ID é utilizado pelas outras tabelas, com exceção da tabela SEASONS que é linkada à tabela RACES pela chave year, indicando o ano da corrida.

Atributos Únicos

Para evitar que haja duplicidade e redundância de dados nas entidades, algumas informações como códigos de identificação ou nomes próprios podem ser inseridas apenas uma vez em cada tabela através do critério UNIQUE, apesar de não servirem como chave primária. Tal critério foi utilizado em AIRPORT em seu code IATA de identificação, CIRCUITS, DRIVER e RACES para a URL de acesso a cada um, CONSTRUCTORS em sua identificação nominal,

Entidades fortes e entidades fracas

Toda entidade é composta por uma chave primária, que serve como sua identificação própria, e se esta possui apenas é composta apenas pela mesma como forma de identificação, pode ser considerada uma entidade forte, por ser auto suficiente, mas para ajudar a organizar as informações de forma mais eficiente, coerente e clara além de facilitar a manipulação dos dados, muitas tabelas também possuem chaves estrangeiras, que referenciam outras entidades criadas, sendo assim dependente destas, e consideradas entidades fracas. As entidades fracas criadas foram:

- DRIVERSTANDINGS, LAPTICES e PITSTOPS que possuem referência a RACES e DRIVERS;
- QUALIFYING que possui referência a RACES, DRIVERS e CONSTRUCTORS;
- RACES que possui referência a SEASONS, e CIRCUITS;
- RESULTS que possui referência a RACES, DRIVERS, CONSTRUCTOS e STATUS

Todas estas entidades referem outras entidades relacionadas às corridas, seus participantes e seus resultados. Ao se criar tabelas únicas que podem ser acessadas por todas as demais entidades auxiliam bastante o trabalho de manter tais dados corretamente preenchidos.