

Trabalho 1

Relatório

SCC-541 Laboratório de Bases de Dados

Leonardo Gonçalves Chahud - 5266649
Lucas Veiga - 9875952
Murilo Franchi - 9790760
Rafael Dantas - 12563686

Prof. Dr Caetano Traina Jr.
PAE: Igor Alberte R. Eleutério

Criação das tabelas

A seguir serão explicadas o porquê de certas decisões terem sido tomadas para cada uma das tabelas do banco de dados. Em cada tabela, usamos o tipo de dado que melhor se encaixa com o dado (Integer para inteiros e Ids, varchar para texto, date para datas e tempos, etc), e explicaremos algumas das decisões tomadas.

Tabela countries

Nesta tabela, não houve dúvidas por parte do grupo. A criação dos atributos foi consistente com o tipo de dado de entrada nos arquivos csv.

- id integer,
- code varchar(2),
- name varchar(255) NOT NULL,
- continent varchar(2) NOT NULL,
- wikipedia_link text NOT NULL,
- keywords text,
- CONSTRAINT pk_countries PRIMARY KEY(id),
- CONSTRAINT uk_countries UNIQUE(code)

Tabela cities

Nesta tabela, não houve dúvidas por parte do grupo. A criação dos atributos foi consistente com o tipo de dado de entrada nos arquivos csv.

- id integer NOT NULL,
- name varchar(200) NOT NULL,
- ascii_name varchar(200) NOT NULL,
- alternate_names text,
- latitude numeric(8,4) NOT NULL,
- longitude numeric(8,4) NOT NULL,
- feature_class varchar(1) NOT NULL,
- feature_code varchar(10) NOT NULL,
- country varchar(2) NOT NULL,
- cc2 varchar(200),
- admin1_code varchar(20),
- admin2_code varchar(80),
- admin3_code varchar(20),
- admin4_code varchar(20),
- population BIGINT NOT NULL,
- elevation integer,
- dem integer,
- time_zone varchar(40) NOT NULL,
- modification DATE NOT NULL,
- CONSTRAINT pk_cities PRIMARY KEY(id)

Tabela airports

Nesta tabela, não houve dúvidas por parte do grupo. A criação dos atributos foi consistente com o tipo de dado de entrada nos arquivos csv.

- id integer NOT NULL,
- ident varchar(20) NOT NULL,
- type varchar(255) NOT NULL,
- name varchar(255) NOT NULL,
- latitude numeric(8,4) NOT NULL,
- longitude numeric(8,4) NOT NULL,
- elevation integer,
- continent varchar(2) NOT NULL,
- iso_country varchar(2) NOT NULL,
- iso_region varchar(25) NOT NULL,
- municipality varchar(255),
- scheduled_service boolean NOT NULL,
- gps_code varchar(20),
- iata_code varchar(20),
- local_code varchar(20),
- home_link text,
- wikipedia_link text,
- keywords varchar(555),
- CONSTRAINT pk_airports PRIMARY KEY(id),
- CONSTRAINT uk_airports UNIQUE(ident)

Tabela circuits

Nesta tabela, não houve dúvidas por parte do grupo. A criação dos atributos foi consistente com o tipo de dado de entrada nos arquivos csv.

- id integer NOT NULL,
- circuit_ref varchar(255) NOT NULL,
- name varchar(255) NOT NULL,
- location varchar(255) NOT NULL,
- country varchar(255) NOT NULL,
- latitude numeric(8,4) NOT NULL,
- longitude numeric(8,4) NOT NULL,
- altitude integer,
- url text NOT NULL,
- CONSTRAINT pk_circuits PRIMARY KEY(id),
- CONSTRAINT uk_circuits UNIQUE(circuit_ref)

Tabela seasons

Nesta tabela, não houve dúvidas por parte do grupo. A criação dos atributos foi consistente com o tipo de dado de entrada nos arquivos csv.

- year integer NOT NULL
- url text NOT NULL
- CONSTRAINT pk_seasons PRIMARY KEY(year)

Tabela status

Nesta tabela, não houve dúvidas por parte do grupo. A criação dos atributos foi consistente com o tipo de dado de entrada nos arquivos csv.

- id integer NOT NULL,
- status varchar(255) NOT NULL,
- CONSTRAINT pk_status PRIMARY KEY(id)

Tabela constructors

Nesta tabela, houve apenas a adição do nome de referência do construtor como chave secundária, consideração essa que foi entendida pelo grupo como possível.

- id integer NOT NULL,
- constructors_ref varchar(255) NOT NULL,
- name varchar(255) NOT NULL,
- nationality varchar(255) NOT NULL,
- url text NOT NULL,
- CONSTRAINT pk_constructors PRIMARY KEY(id),
- CONSTRAINT uk_constructors UNIQUE(constructors_ref)

Tabela drivers

Nesta tabela, houve apenas a adição do nome de referência dos motoristas como chave secundária, consideração essa que foi entendida pelo grupo como possível.

- id integer NOT NULL,
- driver_ref varchar(255) NOT NULL,
- number smallint DEFAULT NULL,
- code varchar(3) DEFAULT NULL,
- forename varchar(255) NOT NULL,
- surname varchar(255) NOT NULL,
- date_of_birth date,
- nationality varchar(255),
- url text NOT NULL,
- CONSTRAINT pk_drivers PRIMARY KEY(id),
- CONSTRAINT uk_drivers UNIQUE(driver_ref).

Tabela races

Os atributos nesta tabela, indicados com comentário, NÃO fazem parte da tabela original. Eles foram adicionados somente para fazermos a importação dos dados sem maiores erros e complicações. No script de inserção dos dados do banco, fazemos o DROP dessas colunas para chegarmos ao formato correto especificado no documento de mapeamento. Quanto ao CASCADE, o grupo entendeu que se houver a deleção de uma temporada pelo administrador do banco de dados, então todas as corridas que ocorreram naquela temporada também poderão ser deletadas.

- id integer NOT NULL,
- year integer NOT NULL,
- round SMALLINT NOT NULL,
- circuit_id integer NOT NULL,
- name varchar(255) NOT NULL,
- date date NOT NULL,
- time time,
- url text,
- fp1_date varchar(10), --Coluna extra para realizar inserção de dados
- fp1_time varchar(10), --Coluna extra para realizar inserção de dados
- fp2_date varchar(10), --Coluna extra para realizar inserção de dados
- fp2_time varchar(10), --Coluna extra para realizar inserção de dados
- fp3_date varchar(10), --Coluna extra para realizar inserção de dados
- fp3_time varchar(10), --Coluna extra para realizar inserção de dados
- quali_date varchar(10), --Coluna extra para realizar inserção de dados
- quali_time varchar(10), --Coluna extra para realizar inserção de dados
- sprint_date varchar(10), --Coluna extra para realizar inserção de dados
- sprint_time varchar(10), --Coluna extra para realizar inserção de dados
- CONSTRAINT pk_races PRIMARY KEY(id),
- CONSTRAINT fk1_races FOREIGN KEY(year) REFERENCES seasons(year) ON DELETE CASCADE,
- CONSTRAINT fk2_races FOREIGN KEY(circuit_id) REFERENCES circuits(id)

Tabela driver_standings

O único ponto a comentar aqui é sobre o CASCADE, o grupo entendeu que se um motorista for deletado da base, então seu driver_standing também deverá ser deletado.

- id integer NOT NULL,
- race_id integer NOT NULL,
- driver_id integer NOT NULL,
- points numeric(8,4) NOT NULL,
- position integer DEFAULT NULL,
- position_text varchar(255),
- wins integer NOT NULL,
- CONSTRAINT pk_driver_standings PRIMARY KEY(id),
- CONSTRAINT fk1_driver_standings FOREIGN KEY(race_id) REFERENCES races(id) ON DELETE CASCADE,
- CONSTRAINT fk2_driver_standings FOREIGN KEY(driver_id) REFERENCES drivers(id) ON DELETE CASCADE

Tabela results

Como a tabela resultados armazena semanticamente os resultados de um participante na corrida, concluímos que se houver a deleção desse participante então seus dados referente aos resultados dele também devem ser apagados. O mesmo se aplica para corridas, caso uma corrida seja deletada então os resultados que se referem a ela também serão apagados.

- id integer NOT NULL,
- race_id integer NOT NULL,
- driver_id integer NOT NULL,
- constructor_id integer NOT NULL,
- number smallint,
- grid smallint NOT NULL,
- position smallint DEFAULT NULL,
- position_text varchar(4) NOT NULL,
- position_order numeric(8,4) NOT NULL,
- points numeric(8,4) NOT NULL,
- laps smallint NOT NULL,
- time varchar(15),
- milliseconds integer,
- fastest_lap smallint,
- rank smallint,
- fastest_lap_time time default '00:00:00.000',
- fastestLapSpeed numeric(6,3),
- status_id integer NOT NULL,
- CONSTRAINT pk_results PRIMARY KEY(id),
- CONSTRAINT fk1_results FOREIGN KEY(race_id) REFERENCES races(id) ON DELETE CASCADE,
- CONSTRAINT fk2_results FOREIGN KEY(driver_id) REFERENCES drivers(id) ON DELETE CASCADE,
- CONSTRAINT fk3_results FOREIGN KEY(constructor_id) REFERENCES constructors(id),
- CONSTRAINT fk4_results FOREIGN KEY(status_id) REFERENCES status(id)

Tabela qualifying

Foi utilizado o tipo de dado varchar para o tempo de qualificação (q1, q2, q3) devido a formatação do dado no arquivo qualifying.csv. Como não existe uma qualificação sem corrida ou piloto, quando essa corrida ou piloto forem apagados, a qualificação também deverá ser apagada.

- id integer NOT NULL,
- race_id integer NOT NULL,
- driver_id integer NOT NULL,
- constructor_id integer NOT NULL,
- number integer NOT NULL,
- position integer,
- q1 varchar(255),
- q2 varchar(255) default NULL,
- q3 varchar(255) default NULL,
- CONSTRAINT pk_qualifying PRIMARY KEY(id),
- CONSTRAINT fk1_qualifying FOREIGN KEY(race_id) REFERENCES races(id) ON DELETE CASCADE,
- CONSTRAINT fk2_qualifying FOREIGN KEY(driver_id) REFERENCES drivers(id) ON DELETE CASCADE,
- CONSTRAINT fk3_qualifying FOREIGN KEY(constructor_id) REFERENCES constructors(id)

Tabela lap_time

Como a tabela **lap_time** armazena dados referentes a volta de um participante na corrida, concluímos que se houver a deleção desse participante então seus dados referente às voltas dele também devem ser apagados. O mesmo se aplica para corridas, caso uma corrida seja deletada então os dados que se referem às voltas dela também serão apagados.

- race_id integer NOT NULL,
- driver_id integer NOT NULL,
- lap smallint NOT NULL,
- position smallint NOT NULL,
- time time NOT NULL,
- milliseconds integer NOT NULL,
- CONSTRAINT pk_lap_time PRIMARY KEY(race_id, driver_id, lap),
- CONSTRAINT fk1_lap_time FOREIGN KEY(race_id) REFERENCES races(id) ON DELETE CASCADE,
- CONSTRAINT fk2_lap_time FOREIGN KEY(driver_id) REFERENCES drivers(id) ON DELETE CASCADE

Tabela **pit_stop**

Como a tabela **pit_stop** armazena dados referentes aos pitstops de um participante na corrida, concluímos que se houver a deleção desse participante então seus dados referente aos pitstops também devem ser apagados. O mesmo se aplica para corridas, caso uma corrida seja deletada então os dados que se referem aos pitstops dela também serão apagados.

- race_id integer NOT NULL,
- race_id integer NOT NULL,
- driver_id integer NOT NULL,
- stop smallint NOT NULL,
- position smallint NOT NULL,
- time time NOT NULL,
- duration varchar(10),
- milliseconds integer,
- CONSTRAINT pk_pit_stop PRIMARY KEY(race_id, driver_id, stop),
- CONSTRAINT fk1_pit_stop FOREIGN KEY(race_id) REFERENCES races(id) ON DELETE CASCADE,
- CONSTRAINT fk2_pit_stop FOREIGN KEY(driver_id) REFERENCES drivers(id) ON DELETE CASCADE

Inserção dos dados

Para a inserção dos dados nas tabelas, foi utilizado o comando COPY do postgres, para importar os dados dos arquivos csv e tsv. Nas tabelas onde existiam elementos Nulos, verificamos qual era esse elemento, em geral '\N', e indicamos no COPY. Como utilizamos o formato CSV por padrão, o delimitador é o ',', exceto na tabela de Cidade, onde o formato é TSV, cujo delimitador é o caractere TAB com o delimitado E'\t'. Após o final das inserções, na tabela races, as colunas fp1_date, fp1_time, fp2_date, fp2_time, fp3_date, fp3_time, quali_date, quali_time, sprint_date, sprint_time serão excluídas por não fazerem parte do modelo relacional fornecido.

airports

- COPY airports
- FROM 'airports.csv'
- WITH (DELIMITER ',', HEADER true, FORMAT CSV);

circuits

- COPY circuits
- FROM 'circuits.csv'
- WITH (DELIMITER ',', NULL '\N', HEADER true, FORMAT CSV);

seasons

- COPY seasons
- FROM 'seasons.csv'
- WITH (DELIMITER ',', NULL '\N', HEADER true, FORMAT CSV);

races

- COPY races
- FROM 'races.csv'
- WITH (DELIMITER ',', NULL '\N', HEADER true, FORMAT CSV);

cities

- COPY cities
- FROM 'Cities15000.tsv'
- WITH (DELIMITER E'\t', HEADER false, FORMAT CSV);

constructors

- COPY constructors
- FROM 'constructors.csv'
- WITH (DELIMITER ',', NULL '\N', HEADER true, FORMAT CSV);

countries

- COPY countries
- FROM 'countries.csv'
- WITH (DELIMITER ',', NULL '\N', HEADER true, FORMAT CSV);

drivers

- COPY drivers
- FROM 'drivers.csv'
- WITH (DELIMITER ',', NULL '\N', HEADER true, FORMAT CSV);

driver_standings

- COPY driver_standings
- FROM 'driver_standings.csv'
- WITH (DELIMITER ',', NULL '\N', HEADER true, FORMAT CSV);

lap_time

- COPY lap_time
- FROM 'lap_times.csv'
- WITH (DELIMITER ',', NULL '\N', HEADER true, FORMAT CSV);

pit_stop

- COPY pit_stop
- FROM 'pit_stops.csv'
- WITH (DELIMITER ',', NULL '\N', HEADER true, FORMAT CSV);

qualifying

- COPY qualifying
- FROM 'qualifying.csv'
- WITH (DELIMITER ',', NULL '\N', HEADER true, FORMAT CSV);

status

- COPY status
- FROM 'status.csv'
- WITH (DELIMITER ',', NULL '\N', HEADER true, FORMAT CSV);

results

- COPY results
- FROM 'results.csv'
- WITH (DELIMITER ',', NULL '\N', HEADER true, FORMAT CSV);