Marca:

* Não pode haver duas marcas com o mesmo ID
* idMarca PRIMARY KEY
* Todas as marcas têm de ter um nome associado
* nome NOT NULL
* O ano de ingresso de uma marca deve ser maior ou igual ao ano da primeira temporada da competição (1949)
* anoDeIngresso CHECK (anoDeIngresso >= 1949)
* O ID da nacionalidade de uma marca deve corresponder ao ID de uma nacionalidade, e, caso seja apagado deve gerar um erro. Por outro lado, caso seja atualizado deverão também o ser todos os atributos que o referenciam.
* idNacionalidade REFERENCES Nacionalidade(idNacionalidade) ON DELETE RESTRICT ON UPDATE CASCADE

Equipa:

* Não pode haver duas equipas com o mesmo ID
* idEquipa PRIMARY KEY
* Todas as equipas têm de ter um nome associado
* nome NOT NULL
* Todas as equipas têm de ter um tipo de equipa que deve ser [de] satélite ou [de] fábrica.
* tipo TEXT CONSTRAINT nn\_equipa\_tipo NOT NULL CONSTRAINT check\_equipa\_tipo CHECK (tipo = "FABRICA" OR tipo = "SATELITE")
* O ID da marca de uma marca deve corresponder ao ID de uma marca, e, caso seja apagado ou atualizado, deverão também o ser todos os atributos que o referenciam.
* idMarca REFERENCES Marca(idmarca) ON DELETE CASCADE ON UPDATE CASCADE

Colaborador:

* Não pode haver dois colaboradores com o mesmo ID
* idColaborador PRIMARY KEY
* Todos os colaboradores têm de ter um nome associado
* nome NOT NULL
* Todos os colaboradores têm de ter uma data de nascimento
* dataDeNascimento NOT NULL
* O ID da nacionalidade de um colaborador deve corresponder ao ID de uma nacionalidade, e, caso seja apagado deve gerar um erro. Por outro lado, caso seja atualizado deverão também o ser todos os atributos que o referenciam.
* idNacionalidade REFERENCES Nacionalidade(idNacionalidade) ON DELETE RESTRICT ON UPDATE CASCADE

Engenheiro:

* Não pode haver dois engenheiros com o mesmo ID, correspondendo este a um ID de um colaborador, e ,caso seja apagado ou atualizado, deverão também o ser todos os atributos que o referenciam.
* idColaborador PRIMARY KEY ON DELETE CASCADE ON UPDATE CASCADE
* Todos os engenheiros têm de ter um ramo no qual estão especializados
* ramo NOT NULL
* O ID da equipa de um engenheiro deve corresponder ao ID de uma equipa, e, caso seja apagado ou atualizado, deverão também o ser todos os atributos que o referenciam.
* idEquipa REFERENCES Equipa(idEquipa) ON DELETE CASCADE ON UPDATE CASCADE

Piloto:

* Não pode haver dois pilotos com o mesmo ID, correspondendo este a um ID de um colaborador, e caso seja apagado ou atualizado, deverão também o ser todos os atributos que o referenciam.
* idColaborador PRIMARY KEY ON DELETE CASCADE ON UPDATE CASCADE
* Todos os pilotos têm de ter um número associado
* numero NOT NULL
* Todos os pilotos têm de ter um número de pontos associado
* numeroPontos NOT NULL
* Todos os pilotos têm de ter um tipo de piloto que deve ser principal ou [de] reserva.
* tipo TEXT CONSTRAINT nn\_piloto\_tipo NOT NULL CONSTRAINT check\_piloto\_tipo CHECK(tipo = "PRINCIPAL" OR tipo = "RESERVA")
* O ID da equipa de um piloto deve corresponder ao ID de uma equipa, e, caso seja apagado ou atualizado, deverão também o ser todos os atributos que o referenciam.
* idEquipa REFERENCES Equipa(idEquipa) ON DELETE CASCADE ON UPDATE CASCADE

Mota:

* Não pode haver duas motas com o mesmo ID
* idMota PRIMARY KEY
* Todas as motas têm de ter uma potência associada, que nunca pode ser nula ou negativa
* potencia NOT NULL CHECK(potencia>0)
* Todas as motas têm de ter um peso, que nunca pode ser nulo ou negativo
* peso NOT NULL CHECK(peso>0)
* Todas as motas têm de ter uma velocidade máxima atingida, que nunca pode ser maior que 340 km/h
* velocidadeMax NOT NULL CHECK(velocidadeMax <= 340)
* O ID da equipa de uma mota deve corresponder ao ID de uma equipa, e, caso seja apagado ou atualizado, deverão também o ser todos os atributos que o referenciam.
* idEquipa REFERENCES Equipa(idEquipa) ON DELETE CASCADE ON UPDATE CASCADE

Pneu:

* Não pode haver dois pneus com o mesmo ID
* idPneu PRIMARY KEY
* Todos os pneus têm de ter uma marca
* marca NOT NULL
* Todos os pneus têm de ter uma rigidez própria, podendo estes ser macio, médio ou duro
* rigidez NOT NULL CHECK(rigidez = "MACIO" OR rigidez = "MEDIO" OR rigidez = "DURO")
* Todos os pneus têm de ter um tipo, podendo estes ser traseiros ou dianteiros
* tipo NOT NULL CHECK(tipo = "TRASEIRO" OR tipo = "DIANTEIRO")
* O ID da mota de um pneu deve corresponder ao ID de uma mota, e, caso seja apagado deve ser substituído por NULL. Por outro lado, caso seja atualizado deverão também o ser todos os atributos que o referenciam.
* idMota REFERENCES Mota(idMota) ON DELETE SET NULL ON UPDATE CASCADE

Corrida:

* Não pode haver duas corridas com o mesmo ID
* idCorrida PRIMARY KEY
* Todas as corridas têm de ser nomeadas
* nome NOT NULL
* Todas as corridas têm de ter uma data associada
* data NOT NULL
* Todas as corridas têm de ter um número de voltas positivo
* numVoltas NOT NULL CHECK(numVoltas > 0)
* O ID do circuito de uma corrida deve corresponder ao ID de um circuito, e, caso seja apagado ou atualizado, deverão também o ser todos os atributos que o referenciam.
* idCircuito REFERENCES Circuito(idCircuito) ON DELETE CASCADE ON UPDATE CASCADE

Circuito:

* Não pode haver dois circuitos com o mesmo ID
* idCircuito PRIMARY KEY
* Todos os circuitos têm de ser nomeados
* nome NOT NULL
* Todos os circuitos têm de estar associados a um país
* pais NOT NULL
* Todos os circuitos têm de ter um perímetro positivo
* perimetro NOT NULL CONSTRAINT CHECK(perimetro > 0)

Grid:

* Não pode haver duas grids com o mesmo ID de corrida e de piloto
* PRIMARY KEY(idCorrida, idPiloto)
* O ID da corrida de uma grid deve corresponder ao ID de uma corrida e, caso seja apagado ou atualizado, deverão também o ser todos os atributos que o referenciam.
* idCorrida REFERENCES Corrida(idCorrida) ON DELETE CASCADE ON UPDATE CASCADE
* O ID do piloto de uma grid deve corresponder ao ID de um piloto e, caso seja apagado ou atualizado, deverão também o ser todos os atributos que o referenciam.
* idPiloto REFERENCES Piloto(idPiloto) ON DELETE CASCADE ON UPDATE CASCADE
* Todos as grids têm de ter uma posição inicial
* posicaoInicial NOT NULL
* Todos as grids têm de ter uma posição final
* posicaoFinal NOT NULL

Classificação Geral:

* Não pode haver duas classificações gerais associadas à mesma época, correspondendo esta a um ano sempre maior que 1949
* epoca PRIMARY KEY CHECK(epoca >= 1949)

Evento:

* Não pode haver dois eventos com o mesmo ID
* idEvento PRIMARY KEY
* Todos os eventos têm de ocorrer durante uma volta com número positivo
* volta NOT NULL CONSTRAINT check\_evento\_volta CHECK(volta > 0)
* O ID da corrida de um evento deve corresponder ao ID de uma corrida e, caso seja apagado ou atualizado, deverão também o ser todos os atributos que o referenciam.
* idCorrida REFERENCES Corrida(idCorrida) ON DELETE CASCADE ON UPDATE CASCADE

Ultrapassagem:

* Não pode haver duas ultrapassagens com o mesmo ID do evento, correspondendo este a um ID de um evento, e, caso seja apagado ou atualizado, deverão também o ser todos os atributos que o referenciam.
* idEvento PRIMARY KEY REFERENCES Evento(idEvento) ON DELETE CASCADE ON UPDATE CASCADE
* Todas as ultrapassagens têm de ter ocorrido num setor com número positivo
* setor NOT NULL CONSTRAINT check\_ultrapassagem\_setor CHECK(setor > 0)

Acidente:

* Não pode haver dois acidentes com o mesmo ID do evento, correspondendo este a um ID de um evento, e, caso seja apagado ou atualizado, deverão também o ser todos os atributos que o referenciam.
* idEvento PRIMARY KEY REFERENCES Evento(idEvento) ON DELETE CASCADE ON UPDATE CASCADE
* Todos os acidentes têm de ter ocorrido num setor com número positivo
* setor NOT NULL CONSTRAINT check\_ultrapassagem\_setor CHECK(setor > 0)

Outro:

* Não pode haver dois outros tipos de evento com o mesmo ID do evento, correspondendo este a um ID de um evento, e, caso seja apagado ou atualizado, deverão também o ser todos os atributos que o referenciam.
* idEvento PRIMARY KEY REFERENCES Evento(idEvento) ON DELETE CASCADE ON UPDATE CASCADE
* Todos esses outros tipos de evento têm de ter uma descrição associada
* descricao NOT NULL

Bandeira:

* Não pode haver duas bandeiras com o mesmo ID do evento e cor, correspondendo este a um ID de um evento, e, caso seja apagado ou atualizado, deverão também o ser todos os atributos que o referenciam.
* idEvento REFERENCES Evento(idEvento) ON DELETE CASCADE ON UPDATE CASCADE
* Todas as bandeiras têm de ter uma cor
* cor NOT NULL
* Não pode haver um tuplo com a mesma conjugação de idEvento e cor.
  + PRIMARY KEY(idEvento, cor)

PilotoCorrida:

* O ID de um colaborador correspondente ao ID da tabela Piloto, deverá ser apagado ou atualizado, caso o Piloto referenciado também o seja.
  + idColaborador INTEGER CONSTRAINT fk\_pilotocorrida\_idpiloto REFERENCES Piloto(idColaborador) ON DELETE CASCADE ON UPDATE CASCADE
* O ID de uma corrida correspondente ao ID da tabela Corrida, deverá ser apagado ou atualizado, caso a Corrida referenciada também o seja.
  + idCorrida INTEGER CONSTRAINT fk\_pilotocorrida\_idcorrida REFERENCES Corrida(idCorrida) ON DELETE CASCADE ON UPDATE CASCADE
* Não pode haver um tuplo com a mesma conjugação de idColaborador e idCorrida.
  + PRIMARY KEY(idColaborador, idCorrida)

PilotoEvento:

* O ID de um colaborador correspondente ao ID da tabela Piloto, deverá ser apagado ou atualizado, caso o Piloto referenciado também o seja.
  + idColaborador INTEGER CONSTRAINT fk\_pilotoevento\_idpiloto REFERENCES Piloto(idColaborador) ON DELETE CASCADE ON UPDATE CASCADE
* O ID de um evento correspondente ao ID da tabela Evento, deverá ser apagado ou atualizado, caso o Evento referenciado também o seja.
  + idEvento INTEGER CONSTRAINT fk\_pilotoevento\_idevento REFERENCES Evento(idEvento) ON DELETE CASCADE ON UPDATE CASCADE
* Não pode haver um tuplo com a mesma conjugação de idColaborador e idEvento.
  + PRIMARY KEY(idColaborador, idEvento)

Nacionalidade:

* Não pode haver duas nacionalidades com o mesmo ID
* idNacionalidade PRIMARY KEY
* Todas as nacionalidades devem ter um nome atribuídas a elas
* nome NOT NULL