

SQL-LINGUAGEM DE DEFINIÇÃO DE DADOS

1. RED BULL AIR RACE

Pretende-se armazenar informação relativa a uma época da competição Red Bull Air Race. Considere o seguinte esquema relacional.

Team (name, country)

Aircraft (model, horsepower, topspeed, width, height, weight)

Pilot (num, firstname, surname, nationality, birthday,
name -> Team, model -> Aircraft)

Race (location, edition, country, date, gates, eliminations)

Participation (num -> Pilot, [location, edition] -> Race,
trainingtime, trainingpos, trainingpenalty,
qualificationtime, qualificationpos, qualificationpenalty,
eliminationtime, eliminationpos, eliminationpenalty)

Duel (numpilot1 -> Pilot, numpilot2 -> Pilot, [location, edition] -> Race, duetype,
timepilot1, timepilot2, penaltypilot1, penaltypilot2)

Utilizando SQL, crie as tabelas necessárias, incluindo as respetivas regras de integridade, para este esquema relacional. Considere como restrições adicionais:

- Não pode haver duas corridas na mesma data;
- Todas as posições de partida em cada etapa (treino, qualificação e eliminação) têm que ser ≥ 1 ;
- Todos os tempos têm que ser positivos;
- Em cada participação, cada piloto só pode ter um tempo (de treino, de qualificação ou de eliminação) se tiver posição de partida respetiva.

2. OFICINA

Um concessionário de automóveis pretende informatizar o seu serviço de reparações em oficina. Considere o seguinte esquema relacional.

Marca (idMarca, nome)

Modelo (idModelo, nome, idMarca -> Marca)

CodPostal (codPostal1, localidade)

Cliente (idCliente, nome, morada, codPostal1 -> CodPostal, codPostal2, telefone)

Carro (idCarro, matricula, idModelo -> Modelo, idCliente -> Cliente)

Reparacao (idReparacao, dataInicio, dataFim,
idCliente -> Cliente, idCarro -> Carro)

Peca (idPeca, codigo, designacao, custoUnitario, quantidade)

ReparacaoPeca (idReparacao -> Reparacao, idPeca -> Peca, quantidade)

PecaModelo (idPeca -> Peca, idModelo -> Modelo)

Especialidade (idEspecialidade, nome, custoHorario)

Funcionario (idFuncionario, nome, morada, codPostal1 -> CodPostal, codPostal2,
telefone, idEspecialidade -> Especialidade)

FuncionarioReparacao (idFuncionario -> Funcionario,
idReparacao -> Reparacao, numHoras)

Utilizando SQL, crie as tabelas necessárias, incluindo as respetivas regras de integridade, para este esquema relacional. Antes de criar as tabelas verifique se as relações obtidas se encontram na Forma Normal de Boyce-Codd. Considere como restrições adicionais:

- Não pode haver quantidades, nem custos nem números de horas negativos;
- Uma reparação não pode terminar antes de começar;
- A matrícula de um carro é única;
- O código de uma peça é único.

3. FACULDADE

Considere a BD das classificações obtidas nas várias provas realizadas pelos alunos nas cadeiras de um ou mais cursos, com as tabelas e instâncias de seguida apresentadas:

ALUNO	
<u>nr</u>	Nome
100	João
110	Manuel
120	Rui
130	Abel
140	Fernando
150	Ismael

PROF	
<u>sigla</u>	Nome
ECO	Eugénio
FNF	Fernando
JLS	João

CADEIRA			
<u>cod</u>	Design	curso	regente
TS1	Teoria dos Sistemas 1	IS	FNF
BD	Bases de Dados	IS	ECO
EIA	Estruturas de Informação e Algoritmos	IS	ECO
EP	Electrónica de Potência	AC	JLS
IE	Instalações Eléctricas	AC	JLS

PROVA			
<u>nr</u>	<u>cod</u>	<u>data</u>	<u>nota</u>
100	TS1	92-02-11	8
100	TS1	93-02-02	11
100	BD	93-02-04	17
100	EIA	92-01-29	16
100	EIA	93-02-02	13
110	EP	92-01-30	12
110	IE	92-02-05	10
110	IE	93-02-01	14
120	TS1	93-01-31	15
120	EP	93-02-04	13
130	BD	93-02-04	12
130	EIA	93-02-02	7
130	TS1	92-02-11	8
140	TS1	93-01-31	10
140	TS1	92-02-11	13
140	EIA	93-02-02	11
150	TS1	92-02-11	10
150	EP	93-02-02	11
150	BD	93-02-04	17
150	EIA	92-01-29	16
150	IE	93-02-02	13

Note que a chave da tabela PROVA é constituída pelos atributos nr, cod e data, permitindo guardar o resultado de mais do que uma prova por cadeira para um mesmo aluno.

Utilizando SQL, crie as tabelas necessárias, incluindo as respetivas regras de integridade, para o modelo relacional dado. Preencha as tabelas com as instâncias apresentadas. [Baseado num exercício de Gabriel David]