U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Equipa Fórmula 1

Turma 13 | Grupo 8

Daniel José Mendes Rodrigues up202006562

Nuno Miguel Lopes da Silva up202005501

Pedro José Ferreira Moreira up201905429

Índice

Contexto da Base de Dados

Introdução

O objetivo deste trabalho é catalogar todos os dados que permitem o pleno funcionamento de uma equipa de Fórmula 1, desde os patrocinadores até ao seu staff.

Departamento

Sobre cada departamento é necessário saber o e-mail administrativo, extensão telefónica e o orçamento.

Em cada departamento podem trabalhar vários membros do staff e certos elementos do staff podem trabalhar em vários departamentos. Dentro de cada departamento um dos membros do Staff é o chefe do departamento.

Infraestrutura/Máquina

Para cada infraestrutura é importante saber que tipo de infraestrutura é (ex: Fábrica, Teste, Pista, Administrativo, …), onde está localizada e a área que ocupa.

Cada edifício tem maquinaria específica (ex.: Túneis de vento, impressoras 3D, …) com determinadas necessidades que permitem o seu pleno funcionamento. Devemos ter em conta a função de cada uma destas máquinas, o seu custo e a data da próxima manutenção, averiguando sempre o seu estado.

Assim, como há um chefe do departamento, também há um chefe para cada infraestrutura responsável pelo seu funcionamento.

Peça

As peças são manufaturadas pelas máquinas e devido à sua diversidade é essencial saber o tipo, quantidade e estado de cada uma das peças.

Ao agregar as peças de diferentes maneiras obtemos diferentes componentes.

Componente

Para cada um dos componentes é importante saber a data de montagem para deduzir a sua durabilidade. Existem vários componentes com diferentes características e atributos, entre eles:

Chassi (rigidez), Pneu (dureza e quilómetros percorridos), Motor (cilindrada), Travões (temperatura máxima, tamanho e pressão), Suspensão (rigidez, anti roll bar, altura), Asa (ângulo) e Transmissão (diferencial e rácios).

Certos componentes podem ser fornecidos por uma equipa parceira ou de um fabricante externo, daí a importância do armazenamento dos principais dados (nome, duração do contrato e pagamento) dos fornecedores.

Staff

Sobre cada elemento do Staff é importante saber as informações pessoais (nome, NIF, género, data de nascimento, nacionalidade, residência), contratuais (data de início de contrato, salário) e os seus contactos (e-mail e telemóvel).

Cada membro do staff tem qualificações diferentes, por isso cada um pode operar máquinas diferentes e devido a obrigações contratuais só pode trabalhar numa equipa.

Equipa/Financiamento

Uma equipa é formada pelos inúmeros membros do staff.

De cada equipa é importante saber o nome, o país de origem e o ano de fundação.

Para que uma equipa seja capaz de competir é fundamental que tenha financiamento, quer seja de investidores ou de patrocinadores. No caso de provir de investidores é crucial saber o nome de cada um e a respetiva participação. Por outro lado, se provir de patrocínios precisamos de saber a marca e os principais dados contratuais (data de início do contrato e a sua duração). Independentemente do tipo de financiamento é fulcral saber o capital investido. É importante apontar que um financiador pode investir em várias equipas.

Carro/Piloto

Cada carro tem um custo de produção associado.

Um carro pode ser conduzido por vários pilotos, mas um piloto só pode conduzir um carro. O piloto é um membro do staff do qual é impreterível ter o conhecimento das suas características físicas, tais como, altura e peso.

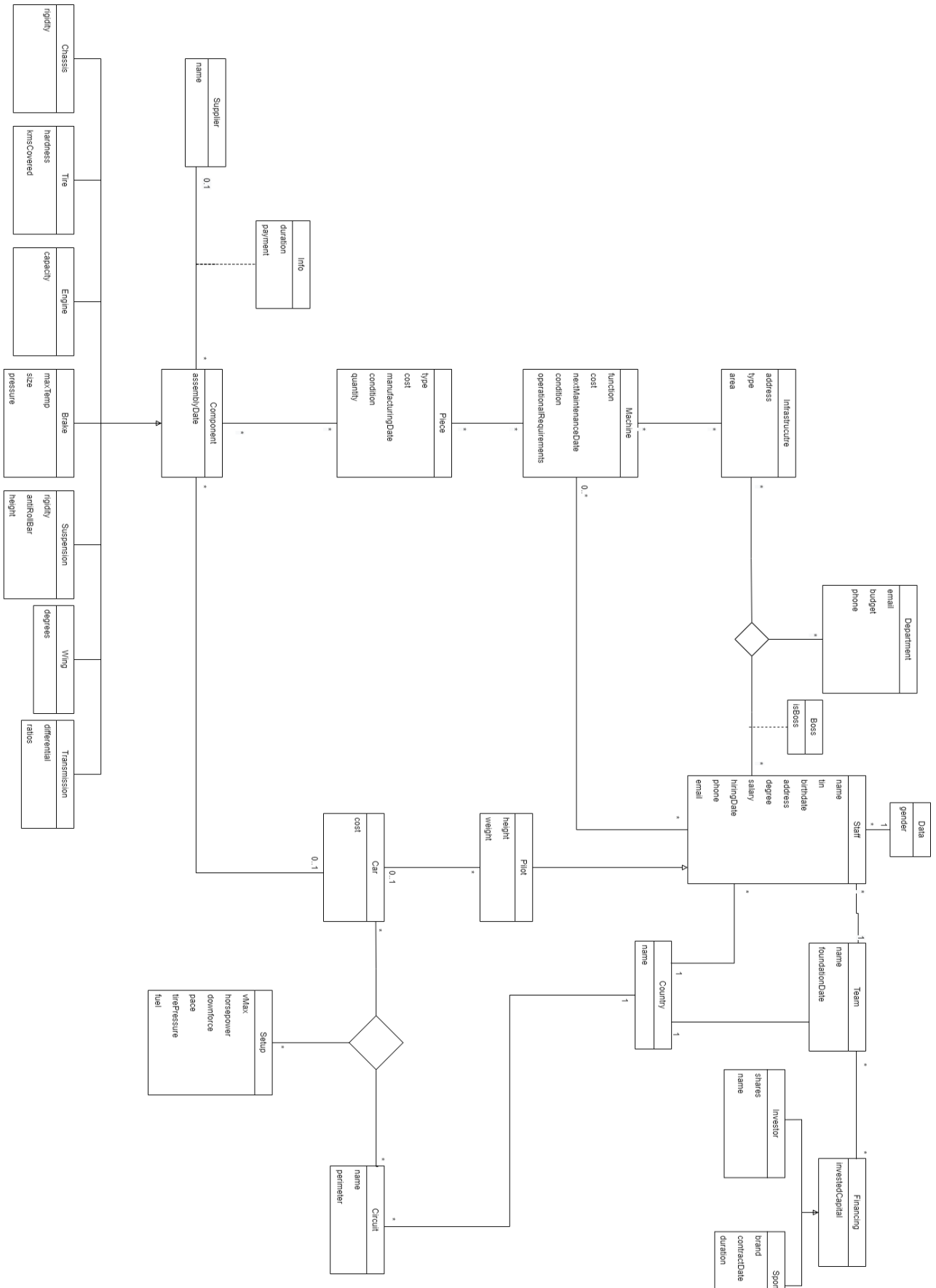
Cada carro é uma agregação dos vários componentes manufaturados pelas máquinas.

Setup/Circuito

O setup de um carro muda de acordo com o circuito onde vai correr. Para tal é necessário saber alguns aspetos fundamentais do carro tais como velocidade máxima, potência, downforce, ritmo, pressão dos pneus e a quantidade de combustível utilizada.

Cada circuito tem um nome, local e perímetro.

Diagrama UML revisto



Esquema Relacional

Department(idDepartment,email, budget, phone)

Staff(idStaff, tin, birthDate, address, degree, salary, hiringDate, phone, email, name -> Country, idTeam -> Team)

Infrastructure(idInfrastructure, address, type, area)

Boss(isBoss)

DepartmentStaffInfrastructureBoss(idDepartment->Department, idStaff->Staff, idInfrastructure->Infrastructure,isBoss->Boss)

Machine(idMachine, function, cost, nextMaintenanceDate, condition, operationRequirements)

InfrastructureMachine(idInfrastructure -> Infrastructure, idMachine -> Machine)

Piece(idPiece, type, cost, manufacturingDate, condition, quantity)

MachinePiece(idMachine -> Machine, idPiece -> Piece)

Supplier(idSupplier, name)

Info(idSupplier->Supplier, idComponent->Component, payment, duration)

ComponentPiece(idComponent -> Component, idPiece -> Piece)

Component(idComponent, assemblyDate, idCar-> Car)

Chassis(idComponent -> Component, rigidity)

Tire(idComponent -> Component, hardness, kmsCovered)

Engine(idComponent -> Component, capacity)

Brake(idComponent -> Component, maxTemp, size, pressure)

Suspension(idComponent -> Component, rigidity, antiRollBar, height)

Wing(idComponent -> Component, degrees)

Transmission(idComponent -> Component, differential, ratios)

Team(idTeam, name, foundationDate, name -> Country)

TeamFinancing(idFinancing->Financing, idTeam -> Team)

Financing(idFinancing, investedCapital)

Investor(idFinancing->Financing, shares, name)

Sponsor(idFinancing->Financing, brand, contractDate, duration)

Pilot(idStaff->Staff, height, weight, idCar -> Car)

Car(idCar, cost)

Circuit(idCircuit, name, perimeter, name -> Country)

Setup(idSetup, vMax, horsepower, downforce, pace, tirePressure, fuel)

CarCircuitSetup(idCar -> Car, idSetup -> Setup, idCircuit -> Circuit)

Análise de Dependências Funcionais e Formas Normais

Supplier
FDS: {idSupplier} -> {name}
{name} -> {idSupplier}
3NF: Sim BCNF: Sim Chave: idSupplier, name

Info
FDS: {idComponent} -> {duration, payment, idSupplier}
3NF: Sim BCNF: Sim Chave: idComponent

Component
FDS: {idComponent} -> {assemblyDate, idCar}
3NF: Sim BCNF: Sim Chave: idComponent

Chassis
FDS: {idComponent} -> {rigidity}
3NF: Sim BCNF: Sim Chave: idComponent

Tire
FDS: {idComponent} -> {hardness, kmsCovered}
3NF: Sim BCNF: Sim Chave: idComponent

Engine
FDS: {idComponent} -> {capacity}
3NF: Sim BCNF: Sim Chave: idComponent

Brake
FDS: {idComponent} -> {maxTemp, size, pressure}
3NF: Sim BCNF: Sim Chave: idComponent

Suspension
FDS: {idComponent} -> {rigidity, antiRollBar, height}
3NF: Sim BCNF: Sim Chave: idComponent

Wing
FDS: {idComponent} -> {degrees}
3NF: Sim BCNF: Sim Chave: idComponent

Transmission
FDS: {idComponent} -> {differential, ratios}
3NF: Sim BCNF: Sim Chave: idComponent

Infrastructure
FDS: {idInfrastructure} -> {location, type, area}
3NF: Sim BCNF: Sim Chave: idInfrastructure

Machine
FDS: {idMachine} -> {function, cost, nextMaintenanceDate, condition, operationalRequirements}
3NF: Sim BCNF: Sim Chave: idMachine

Piece
FDS: {idPiece} -> {type, cost, manufacturingDate, condition, quantity}
3NF: Sim BCNF: Sim Chave: idPiece

Pilot
FDS: {idPilot} -> {Height,Weight}
3NF: Sim BCNF: Sim Chave: idPilot

Car
FDS: {idCar} -> {Cost}
3NF: Sim BCNF: Sim Chave: idCar

Setup
FDS: {idSetup} -> {vMax, horsepower, downforce, pace, tirePressure, fuel}
3NF: Sim BCNF: Sim Chave: idSetup

Circuit
FDS: {idCircuit} -> {name, location, perimeter}
{name} -> {idCircuit, location perimeter}
3NF: Sim BCNF: Sim Chaves: idCircuit, name

Department
FDS: {idDepartment} -> {email, budget, phone}
{email} -> {idDepartment, budget, phone}
{phone} -> {idDepartment, email, budget}
3NF: Sim BCNF: Chaves: idDepartment, email, phone

Staff
FDS: {idStaff} -> {name, tin, birthDate, address, degree, salary, hiringDate, phone, email}
{tin} -> {name, idStaff, birthDate, address, degree, salary, hiringDate, phone, email}
{email} -> {name, tin, birthDate, address, degree, salary, hiringDate, phone, idStaff}
{phone} -> {name, tin, birthDate, address, degree, salary, hiringDate, email, idStaff}
3NF: Sim BCNF: Sim Chaves: idSatff, tin, email, phone

Team
FDS: {idTeam} -> {name, foundationDate}
{name} -> {idTeam, foundationDate}
3NF: Sim BCNF: Sim Chaves:idTeam, name

Financing
FDS: {idFinancing} -> {investedCapital}
3NF: Sim BCNF: Sim Chave: idFinancing

Investor
FDS: {idFinancing} -> {shares, name}
3NF: Sim BCNF: Sim Chave: idFinancing

Sponsor
FDS: {idSponsor} -> {brand, contractDate, duration}
{brand} -> {idSponsor, contractDate, duration}
3NF: Sim BCNF: Sim Chaves: idSponsor, brand

Justificação

Analisando cada uma das dependências funcionais que constituem uma relação, podemos constatar que o lado esquerdo de cada uma é constituído por uma chave dessa relação sendo condição suficiente para que esta esteja na Forma Normal de Boyce-Codd. Como a Forma Normal de Boyce-Codd está contida na Terceira Forma Normal, podemos concluir que todas as relações respeitam a 3NF.

Restrições

Geral
A maior parte dos atributos das relações não podem ser nulos.
As chaves primárias e estrangeiras das relações já foram apresentadas na análise das dependências funcionais.

Supplier
- Dois fornecedores não podem ter o mesmo nome.
    - name UNIQUE

Info
- É obrigatório que os atributos *duration* e *payment* sejam valores positivos não nulos
    - CHECK(duration > 0 AND payment > 0)

Chassis
- Cada chassi tem um valor associado à sua rigidez, podendo variar entre 0% a 100%
    - CHECK(rigidity >= 0 AND rigidity <= 100)

Tire
- Um pneu pertence a uma das oitos categorias de dureza
    - CHECK(hardness >= 1 AND hardness <= 8)
- O atributo kmsCovered tem de ser maior ou igual a 0
    - CHECK(kmsCovered >= 0)

Engine
- O motor tem uma cilindrada menor ou igual a 1600 cm^3
    - CHECK(capacity <= 1600)

Brake
- Os travões têm uma temperatura máxima positiva
    - CHECK(maxTemp>0)

Suspension
- A rigidez da suspensão e da *antiRollBar* podem ser quantificados por valores numéricos compreendidos entre 1 e 11.
    - CHECK(rigidity >= 1 AND rigidity <= 11 AND antiRollBar >= 1 AND antiRollBar <= 11)
- A altura da suspensão não pode exceder os 1,1 metros
    - CHECK(height >= 0 AND height <= 1,1)

Wing
- A inclinação da asa pode variar entre 5º a 15º
    - CHECK (degrees >= 5 AND degrees <= 15)

Transmission

- Os valores do rácio e do diferencial de uma transmissão estão compreendidos entre 50% e 100%.
    - CHECK(differential>=50 AND differential<= 100 AND ratio >=50 AND ratio <= 100)

Department

- É obrigatório que cada departamento tenha um email e um número de telefone próprio
    - email UNIQUE
    - phone UNIQUE
- É obrigatório que o orçamento seja um valor positivo não nulo
    - CHECK(budget > 0)

Staff

- Cada elemento da Staff deve ter nif, um endereço, um grau de qualificação, um número de telemóvel e um email não nulo.
    - tin phone email UNIQUE
- É obrigatório que o salário seja um valor positivo não nulo
    - CHECK(salary > 0)

Team

- Cada equipa deve ter um nome exclusivo
    - name UNIQUE

Investor

- É necessário que os atributos *investedCapital* e *shares* sejam valores positivos
    - CHECK(investedCapital > 0 AND shares > 0)

Sponsor

- É necessário que os atributos *investedCapital* e *duration* sejam valores positivos
    - CHECK(investedCapital > 0 AND duration > 0)
- Uma marca tem um nome exclusivo
    - brand UNIQUE

Infrastructure

- É necessário que o atributo *area* seja valor positivo
    - CHECK(area > 0)

Machine

- É necessário que os atributos *function, nextMaintenanceDate* e *condition* não sejam valores nulos
    - function NOT NULL
    - nextMaintenanceDate NOT NULL

    - condition NOT NULL
- É necessário que os atributo *nextMaintenanceDate* seja superior a 12-12-2021
    - CHECK(nextMaintenanceDate > 2021-12-12)

Piece

- É necessário que o atributo *condition* não seja nulo
    - condition NOT NULL
- É necessário que o atributo *quantity* seja superior ou igual a 0
    - CHECK(quantity >= 0)

Pilot

- É necessário que os atributos *weight, height* só possa variar entre 30 e 150
    - CHECK(weight >= 30 AND height <= 150)

Car

- É necessário que o atributo *cost* seja positivo
    - CHECK(cost > 0)

Setup

- É necessário que os atributos *vMax, horsepower, tirePressure, fuel* seja positivo
    - CHECK(vMax > 0 AND horsepower > 0 AND tirePressure > 0 AND fuel > 0)

Circuit

- É necessário que o atributo *name* seja único
    - name UNIQUE

O trabalho foi dividido equitativamente pelos membros do grupo.