



ÉPOCA EQUESTRE DE ALTA COMPETIÇÃO

BASEADO NA LONGINES GLOBAL CHAMPIONS TOUR

**Base de Dados**

**2ºAno MIEIC**

8 de março de 2020

Inês Silva, up201806385

Mariana Truta, up201806543

Rita Peixoto, up201806257

Grupo 4, Turma 7

**Índice**

[Definição do Modelo Conceptual 2](#_Toc36400583)

[1. Contexto 2](#_Toc36400584)

[2. Diagrama UML 3](#_Toc36400585)

[Definição do Esquema Relacional 4](#_Toc36400586)

[Análise de Dependências Funcionais e Formas Normais 7](#_Toc36400587)

[Adição de restrições à base de dados 10](#_Toc36400588)

# Definição do Modelo Conceptual

## Contexto

Tendo como base a *Longines Global Champions Tour*, uma época (*Season*) equestre de alta competição é constituída por uma liga individual (*TourCompetition*) e uma liga de equipas (*LeagueCompetition*), sendo esta última constituída por um número de equipas (*Team*) de quatro cavaleiros determinado anualmente.

Ambas as ligas são constituídas pelas mesmas etapas *(Event),* caracterizadas por uma cidade e datas de início e fim.

Cada etapa tem várias provas (*Class*) com um número de participantes, um número de obstáculos e um tempo máximo de conclusão. No final de cada prova, os três primeiros classificados são distinguidos com um prémio monetário (*Prize*), sendo este fixo para todas as etapas, por isso é necessário guardar o número de pontos de cada cavaleiro em cada participação (*Participation*). Caso o cavaleiro não termine uma prova, é preciso indicar o motivo, podendo este ser desistência ou desqualificação.

De cada cavaleiro (*Rider*) pretende-se saber o nome, o número de federado, o país, a data de nascimento, o valor monetário ganho até ao momento e os cavalos com que vai competir ao longo da época. Do cavalo (*Horse*) é necessário saber o respetivo cavaleiro, o ano de nascimento, o ID, a raça, o género e a cor do pelo.

Cada cavaleiro tem a sua equipa constituída por um tratador (*Groom*: nome e data de nascimento) e um treinador (*Coach*: nome, país e data de nascimento), podendo ter ou não patrocinadores (*Sponsor*: id, profissão, NIF e valor do patrocínio). Além disso, o cavaleiro está associado a um estábulo (*Stable*: ano de criação, se é federado ou não, número de cavalos, número de praticantes, número de funcionários e área). É importante realçar que ser federado implica que esteja associado a um clube. Todos os cavaleiros têm um veterinário de urgência (*Vet*), sabendo-se o seu nome, contacto telefónico e preço por consulta.

## Diagrama UML

Uma imagem com texto, mapa

Descrição gerada automaticamente

# Definição do Esquema Relacional

**Coach**(CID, Name, Country, BirthDate)

CID é a chave primária.

**Groom** (GID, Name, BirthDate)

GID é a chave primária.

**Sponsor**(SID, Job, NIF)

SID  é a chave primária.

**Vet**(VID, Name, PhoneNumber, AppointementCost)

VID  é a chave primária.

**Club**(Name, CreationYear)

Name  é a chave primária.

**Stable**(STID, InaugurationYear, NoHorses, NoPractitioners, NoEmployees, Area, CName🡪Club)

STID  é a chave primária.

**Team**(TeamID, Name, AmountGain, TotalPoints)

TeamID  é a chave primária.

AmountGain e TotalPoints são atributos derivados.

**Rider**(RiderID, Name, Country, NoFederated, BirthDate, AmountGain, TotalPoints, CoachID🡪Coach, VetID🡪Vet, GroomID🡪Groom, StableID🡪Stable, TeamID🡪Team)

RiderID  é a chave primária.

CoachID, VetID, GroomID, StableID e TeamID são chaves estrangeiras.

**Sponsors**(RiderID🡪Rider, SponsorID🡪Sponsor)

RiderID e SponsorID são a chave primária composta.

RiderID e SponsorID são ambas chaves estrangeiras

**Horse**(HorseID, BirthDate, Breed, Gender, CoatColor, RiderID🡪Rider)

HorseID é a chave primária.

RiderID é uma chave estrangeira.

**Participation**(DateTime🡪Class, Place, Points, PTime)

DateTime e Place são a chave primária composta.

DateTime é uma chave estrangeira.

**IndividualParticipation**([DateTime, Place]🡪Participation, RiderID🡪Rider, HorseID🡪Horse)

DateTime e Place são a chave primária composta.

{DateTime, Place}, RiderID e HorseID são chaves estrangeiras.

**IndividualDisqualification**([DateTime, Place]🡪Participation, RiderID🡪Rider)

DateTime e Place são a chave primária composta.

{DateTime, Place} e RiderID são chaves estrangeiras.

**IndividualWithdrawal**([DateTime, Place]🡪Participation, RiderID🡪Rider)

DateTime e Place são a chave primária composta.

{DateTime, Place} e RiderID são chaves estrangeiras.

**TeamParticipation**([DateTime, Place]🡪Participation, TeamID🡪Team)

DateTime e Place são a chave primária composta.

{DateTime, Place} e TeamID são chaves estrangeiras.

**HorseParticipation**([DateTime, Place]🡪Participation, HorseID🡪Horse)

DateTime e Place são a chave primária composta.

{DateTime, Place} e HorseID são chaves estrangeiras.

**TeamDisqualification**([DateTime, Place]🡪Participation, TeamID🡪Team)

DateTime e Place são a chave primária composta.

{DateTime, Place} e TeamID são chaves estrangeiras.

**TeamWithdrawal**([DateTime, Place]🡪Participation, TeamID🡪Team)

DateTime e Place são a chave primária composta.

{DateTime, Place} e TeamID são chaves estrangeiras.

**Season(Year)**

Year é a chave primária.

**Event**(City, Year🡪Season, BeginDate, EndDate)

City e Year são a chave primária composta.

Year é uma chave estrangeira.

**Class**(DateTime, [City, Year]🡪Event, MaxTime, NoObstacles)

DateTime é a chave primária.

City e Year são uma chave estrangeira.

**Prize**(Place, Class🡪Class, Value)

Place e Class são a chave primária composta.

Class é uma chave estrangeira.

# Análise de Dependências Funcionais e Formas Normais

Para uma tabela se encontrar na 3ª Forma Normal, tem obrigatoriamente de estar na 2ª Forma Normal o que implica que esteja também na 1ª Forma Normal. Isto é, para além de nenhum atributo poder possuir mais do que um valor e não poder haver grupos de valores repetidos numa relação, para uma relação estar na 3ª Forma Normal é necessário que todos os atributos que não pertençam a qualquer chave candidata dependam inteiramente da chave primária (e não apenas de parte dela) e que não existam dependências funcionais entre os atributos não chave.

No caso de haver relações que ainda transportam anomalias após ter sido aplicada a 3ª Forma Normal, é frequente utilizar-se a Forma Normal de Boyce-Codd (BCNF) que requer que não existam dependências funcionais não triviais que não dependam exclusivamente de uma chave candidata.

**Coach**(CID, Name, Country, BirthDate)

CID 🡪 Name, Country, BirthDate

**Groom** (GID, Name, BirthDate)

GID 🡪 Name, BirthDate

**Sponsor**(SID, Job, NIF)

SID 🡪 Job, NIF

NIF 🡪 SID

A segunda dependência funcional viola a Forma Normal de Boyce-Codd já que não existe nenhuma chave do lado esquerdo da mesma. Por outro lado, analisando a closure de NIF “{NIF}+={NIF, SID}”, não estão incluídos todos os atributos da tabela e por isso é violada a BCNF.

**Vet**(VID, Name, PhoneNumber, AppointementCost)

VID 🡪 Name, PhoneNumber, AppointementCost

PhoneNumber 🡪 VID

A segunda dependência funcional viola a Forma Normal de Boyce-Codd já que não existe nenhuma chave do lado esquerdo da mesma. Por outro lado, analisando a closure de PhoneNumber “{PhoneNumber}+={PhoneNumber, VID}”, não estão incluídos todos os atributos da tabela e por isso é violada a BCNF.

**Club**(Name, Creation Year)

Name 🡪 CreationYear

**Stable**(STID, InaugurationYear, NoHorses, NoPractitioners, NoEmployees, Area, CName🡪Club )

STID 🡪 InaugurationYear, NoHorses, NoPractitioners, NoEmployees, Area, CName

**Team**(TeamID, Name, AmountGain, TotalPoints)

TeamID 🡪 Name, AmountGain,TotalPoints

Name 🡪 TeamID

A segunda dependência funcional viola a Forma Normal de Boyce-Codd já que não existe nenhuma chave do lado esquerdo da mesma. Por outro lado, analisando a closure de Name “{Name}+={ Name, TeamID}”, não estão incluídos todos os atributos da tabela e por isso é violada a BCNF.

**Rider**(RiderID, Name, Country, NoFederated, BirthDate, AmountGain, TotalPoints, CoachID🡪Coach, VetID🡪Vet, GroomID🡪Groom, StableID🡪Stable, TeamID🡪Team)

RiderID 🡪 Name, Country, NoFederated, BirthDate, AmountGain, TotalPoints, CoachID, VetID, GroomID, StableID, TeamID

**Horse**(HorseID, BirthDate, Breed, Gender, CoatColor, RiderID🡪Rider)

HorseID 🡪 BirthDate, Breed, Gender, CoatColor, RiderID

**Participation**(DateTime 🡪Class, Place, Points, PTime)

DateTime, Place 🡪 Points, PTime

Points, PTime 🡪 Place

A segunda dependência funcional viola a Forma Normal de Boyce-Codd já que não existe nenhuma chave do lado esquerdo da mesma. Por outro lado, analisando a closure “{Points, PTime}+={Points, PTime, Place}”, não estão incluídos todos os atributos da tabela e por isso é violada a BCNF.

**IndividualParticipation**([DateTime, Place]🡪Participation, RiderID🡪Rider, HorseID🡪Horse)

DateTime, Place 🡪 RiderID, HorseID

**IndividualDisqualification**([DateTime, Place]🡪Participation, RiderID🡪Rider)

DateTime, Place 🡪 RiderID

**IndividualWithdrawal**([DateTime, Place]🡪Participation, RiderID🡪Rider)

DateTime, Place 🡪 RiderID

**TeamParticipation**([DateTime, Place]🡪Participation, TeamID🡪Team)

DateTime, Place 🡪 TeamID

**TeamDisqualification**([DateTime, Place]🡪Participation,TeamID🡪Team)

DateTime, Place 🡪 TeamID

**TeamWithdrawal**([DateTime, Place]🡪Participation, TeamID🡪Team)

DateTime, Place 🡪 TeamID

**Event**(City, Year🡪Season, BeginDate, EndDate)

City, Year 🡪 BeginDate, EndDate

BeginDate, EndDate 🡪 Year, City

**Class**(DateTime, [City, Year]🡪Event, MaxTime, NoObstacles)

DateTime 🡪 City, Year, MaxTime, NoObstacles

**Prize**(Place, Class🡪Class, Value)

Place, Class 🡪 Value

Value, Class 🡪 Place

Pode-se afirmar que nenhuma relação apresentada anteriormente viola a 3ª Forma Normal já que todos os atributos não chave dependem inteira e exclusivamente da totalidade da chave. Além disso, para além das violações da BCNF mencionadas, todas as outras relações estão na Forma Normal de Boyce-Codd uma vez que do lado esquerdo de cada dependência funcional existe sempre uma chave.

# Adição de restrições à base de dados

De forma a limitar o tipo de dados que podem ser armazenados numa tabela, é necessário definir algumas restrições nas colunas ou tabelas. Assim, caso o utilizador tente armazenar dados que não obedeçam a essas restrições, dá origem a um erro. São normalmente utilizadas 5 restrições, sendo estas a restrição chave (PRIMARY KEY), a restrição de integridade referencial (FOREIGN KEY), a restrição UNIQUE, a restrição CHECK e a restrição NOT NULL.

**Coach**

* CID é a chave primária (restrição PRIMARY KEY), isto é, não pode ser nula e tem de ser única;
* CName e Country não podem ser nulos (restrição NOT NULL).

**Groom**

* GID é a chave primária (restrição PRIMARY KEY), isto é, não pode ser nula e tem de ser única;
* GName não pode ser nulo (restrição NOT NULL).

**Sponsor**

* SID é a chave primária (restrição PRIMARY KEY), isto é, não pode ser nula e tem de ser única;
* Job e NIF não podem ser nulos (restrição NOT NULL);
* NIF tem de ter 9 dígitos (restrição CHECK (length(NIF) = 9)), de ser único (restrição UNIQUE) e maior do que 0 (restrição CHECK NIF > 0).

**Vet**

* VID é a chave primária (restrição PRIMARY KEY), isto é, não pode ser nula e tem de ser única;
* VName, PhoneNumber e AppointementCost não podem ser nulos (restrição NOT NULL);
* AppointementCost e PhoneNumber não podem ser negativos (restrições: CHECK AppointementCost >= 0, CHECK PhoneNumber > 0);
* PhoneNumber tem de ser único (restrição UNIQUE) e tem de ter 9 dígitos (restrição CHECK (length(NIF) = 9)).

**Club**

* CName é a chave primária (restrição PRIMARY KEY), isto é, não pode ser nula e tem de ser única;
* CreationYear não pode ser nulo (restrição NOT NULL) e tem de ser maior do que 0 (restrição CHECK CreationYear > 0).

**Stable**

* STID é a chave primária (restrição PRIMARY KEY), isto é, não pode ser nula e tem de ser única;
* CName é uma chave estrangeira (restrição FOREIGN KEY);
* NoHorses, NoPractitioners, NoEmployees e Area não podem ser NULL (restrição NOT NULL) nem negativos (restrições: CHECK NoHorses >= 0, CHECK NoPractitioners >= 0, CHECK NoEmployees >= 0 e CHECK Area > 0);
* InaugurationYear não pode ser nulo (restrição NOT NULL) e tem de ser maior do que 0 (restrição CHECK InaugurationYear > 0).

**Team**

* TeamID é a chave primária (restrição PRIMARY KEY), isto é, não pode ser nula e tem de ser única;
* TName tem de ser único (restrição UNIQUE) e não pode ser nulo (restrição NOT NULL);
* AmountGain e TotalPoints não podem ser nulos (restrição NOT NULL) e têm de ser maiores ou iguais a 0 (restrições: CHECK AmountGain >= 0, CHECK TotalPoints >= 0).

**Rider**

* RiderID é a chave primária (restrição PRIMARY KEY), isto é, não pode ser nula e tem de ser única;
* NoFederated tem de ser único (restrição UNIQUE), não pode ser nulo (restrição NOT NULL) e tem de ser maior do que 0 (restrição CHECK NoFederated > 0);
* AmountGain e TotalPoints não podem ser nulos (restrição NOT NULL) e têm de ser maiores ou iguais a 0 (restrições: CHECK AmountGain >= 0, CHECK TotalPoints >= 0);
* RName , Country, GroomID, VetID, StableID e BirthDate não podem ser nulos (restrição NOT NULL).

**Sponsors**

* RiderID e SponsorID são uma chave primária composta (restrição PRIMARY KEY(RiderID, SponsorID)) e são também chaves estrangeiras (restrição FOREIGN KEY).

**Horse**

* HorseID é a chave primária (restrição PRIMARY KEY), isto é, não pode ser nula e tem de ser única;
* RiderID é uma chave estrangeira (restrição FOREIGN KEY) e não pode ser nulo (restrição NOT NULL);
* Gender só pode ser ‘Male’ ou ‘Female’ (restrição CHECK Gender = "Male" OR Gender = "Female") e não pode ser nulo (restrição NOT NULL);
* CoatColor não pode ser nulo (restrição NOT NULL);
* Há algumas possibilidades para a CoatColor que são mencionadas na seguinte restrição: CHECK (CoatColor = "Gray" OR CoatColor = "Bay" OR CoatColor = "Buckskin" OR CoatColor = "Brown" OR CoatColor = "Black" OR CoatColor = "Dun" OR CoatColor = "Roan" OR CoatColor = "Palomino" OR CoatColor = "Chestnut" OR CoatColor = "Pinto" OR CoatColor = "Paint").

**Participation**

* DateTime e Place são uma chave primária composta (restrição PRIMARY KEY (DateTime, Place));
* DateTime é uma chave estrangeira (restrição FOREIGN KEY);
* Place tem de ser maior do que 0 (restrição CHECK Place > 0);
* Points e PTime têm de ser maiores ou iguais que 0 (restrições CHECK: CHECK Points >= 0, CHECK PTime >= 0) e não podem ser nulos (restrição NOT NULL). Realça-se o facto de, no caso de desistência ou desqualificação, o PTime ser igual 0.

**IndividualParticipation**

* DateTime e Place são a chave primária composta (restrição PRIMARY KEY(DateTime, Place)) e são uma chave estrangeira (restrição FOREIGN KEY(DateTime, Place));
* Place tem de ser maior do que 0 (restrição CHECK Place > 0);
* RiderID e HorseID são chaves estrangeiras (restrição FOREIGN KEY) e não podem ser nulos (restrição NOT NULL).

**IndividualDisqualification**

* DateTime e Place são a chave primária composta (restrição PRIMARY KEY(DateTime, Place)) e são uma chave estrangeira (restrição FOREIGN KEY(DateTime, Place));
* Place tem de ser maior do que 0 (restrição CHECK Place > 0);
* RiderID é uma chave estrangeira (restrição FOREIGN KEY) e não pode ser nulo (restrição NOT NULL).

**IndividualWithdrawal**

* DateTime e Place são a chave primária composta (restrição PRIMARY KEY(DateTime, Place)) e são uma chave estrangeira (restrição FOREIGN KEY(DateTime, Place));
* Place tem de ser maior do que 0 (restrição CHECK Place > 0);
* RiderID é uma chave estrangeira (restrição FOREIGN KEY) e não pode ser nulo (restrição NOT NULL).

**HorseParticipation**

* DateTime, Place e HorseID são a chave primária composta (restrição PRIMARY KEY(DateTime, Place, HorseID));
* DateTime e Place são uma chave estrangeira (restrição FOREIGN KEY(DateTime, Place));
* Place tem de ser maior do que 0 (restrição CHECK Place > 0);
* HorseID é uma chave estrangeira (restrição FOREIGN KEY).

**TeamParticipation**

* DateTime e Place são a chave primária composta (restrição PRIMARY KEY(DateTime, Place)) e são uma chave estrangeira (restrição FOREIGN KEY(DateTime, Place));
* Place tem de ser maior do que 0 (restrição CHECK Place > 0);
* TeamID é uma chave estrangeira (restrição FOREIGN KEY) e não pode ser nulo (restrição NOT NULL).

**TeamDisqualification**

* DateTime e Place são a chave primária composta (restrição PRIMARY KEY(DateTime, Place)) e são uma chave estrangeira (restrição FOREIGN KEY(DateTime, Place));
* Place tem de ser maior do que 0 (restrição CHECK Place > 0);
* TeamID é uma chave estrangeira (restrição FOREIGN KEY) e não pode ser nulo (restrição NOT NULL).

**TeamWithdrawal**

* DateTime e Place são a chave primária composta (restrição PRIMARY KEY(DateTime, Place)) e são uma chave estrangeira (restrição FOREIGN KEY(DateTime, Place));
* Place tem de ser maior do que 0 (restrição CHECK Place > 0);
* TeamID é uma chave estrangeira (restrição FOREIGN KEY) e não pode ser nulo (restrição NOT NULL).

**Season**

* Year é a chave primária (restrição PRIMARY KEY), isto é, não pode ser nula e tem de ser única. Além disso, também tem de ser maior do que 0 (restrição CHECK Year > 0).

**Event**

* City e Year são a chave primária composta (restrição PRIMARY KEY (City, Year));
* Year é uma chave estrangeira (restrição FOREIGN KEY);
* BeginDate e EndDate são datas que não podem ser nulas (restrição NOT NULL) e EndDate tem de ser posterior à BeginDate (restrição CHECK EndDate > BeginDate).

**Class**

* DateTime é a chave primária (restrição PRIMARY KEY), isto é, não pode ser nula e tem de ser única;
* City e Year são uma chave estrangeira (restrição FOREIGN KEY (City, Year));
* Year tem de ser maior do que 0 (restrição CHECK Year > 0);
* MaxTime e NoObstacles são valores positivos (restrições: CHECK MaxTime > 0 e CHECK NoObstacles > 0) e não podem ser nulos (restrição NOT NULL).

**Prize**

* ClassID e Place são a chave primária composta (restrição PRIMARY KEY(ClassID, Place);
* ClassID é uma chave estrangeira (restrição FOREIGN KEY);
* Place só pode tomar três valores (restrição CHECK (Place = 1 OR Place = 2 OR Place = 3));
* Value não pode ser negativo (restrição CHECK Value >= 0) nem nulo (restrição NOT NULL).