

ÉPOCA EQUESTRE DE ALTA COMPETIÇÃO

BASEADO NA LONGINES GLOBAL CHAMPIONS TOUR



Base de Dados

2ºAno MIEIC

5 de abril de 2020

Inês Silva, up201806385
Mariana Truta, up201806543
Rita Peixoto, up201806257
Grupo 4, Turma 7

Índice

Definição do Modelo Conceptual.....	2
1. Contexto	2
2. Diagrama UML	3
Definição do Esquema Relacional.....	4
Análise de Dependências Funcionais e Formas Normais.....	7
Adição de restrições à base de dados.....	11
Interrogação da Base de dados	18
Interrogação 1	18
Interrogação 2	18
Interrogação 3	18
Interrogação 4	18
Interrogação 5	18
Interrogação 6	18
Interrogação 7	18
Interrogação 8	18
Interrogação 9	18
Interrogação 10	18
Adição de gatilhos à base de dados.....	19
Gatilho 1	19
Gatilho 2	19
Gatilho 3.....	19

Definição do Modelo Conceptual

1. 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.

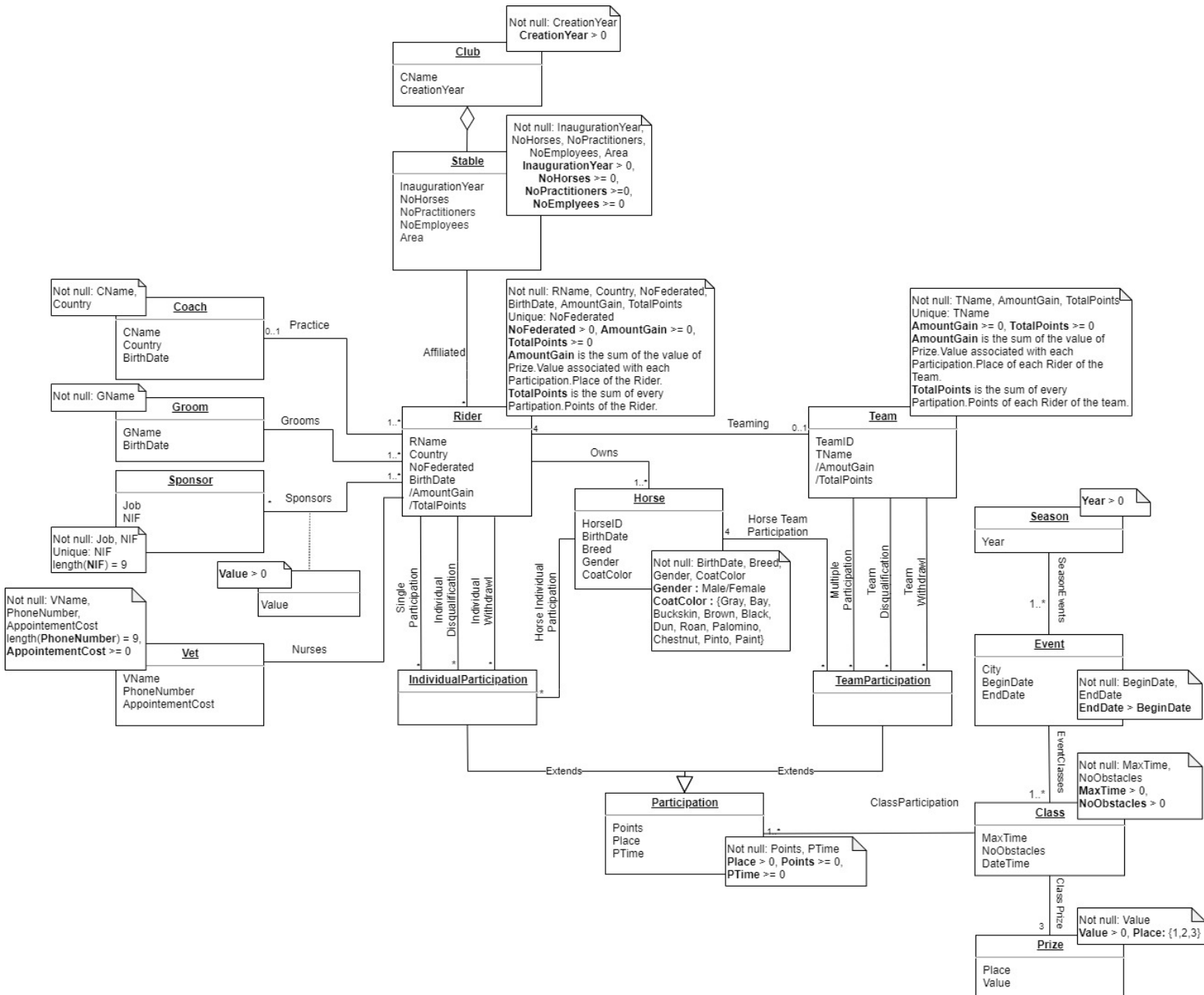
Ambas as ligas são constituídas pelas mesmas etapas (*Event*), caracterizadas por uma cidade e datas de início e fim.

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, número de cavalos, número de praticantes, número de funcionários e área). É importante realçar que, se for federado, o estábulo está associado a um clube. Todos os cavaleiros têm um veterinário de urgência (*Vet*), sabendo-se o seu nome, contacto telefónico e preço por consulta.

2. Diagrama UML



Definição do Esquema Relacional

Coach(CID, CName, Country, BirthDate)

CID é a chave primária.

Groom (GID, GName, BirthDate)

GID é a chave primária.

Sponsor(SID, Job, NIF)

SID é a chave primária.

Vet(VID, VName, PhoneNumber, AppointmentCost)

VID é a chave primária.

Club(CName, CreationYear)

Name é a chave primária.

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

STID é a chave primária.

Team(TeamID, TName, AmountGain, TotalPoints)

TeamID é a chave primária.

AmountGain e TotalPoints são atributos derivados.

Rider(RiderID, RName, 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, Value)

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]→IndividualParticipation, RiderID→Rider, HorseID→Horse)

DateTime e Place são a chave primária composta.

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

IndividualDisqualification([DateTime, Place]→IndividualParticipation, RiderID→Rider)

DateTime e Place são a chave primária composta.

{DateTime, Place} e RiderID são chaves estrangeiras.

IndividualWithdrawal([DateTime, Place]→IndividualParticipation, 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]→TeamParticipation, HorseID→Horse)

DateTime e Place são a chave primária composta.

{DateTime, Place} e HorseID são chaves estrangeiras.

TeamDisqualification([DateTime, Place]→TeamParticipation, TeamID→Team)

DateTime e Place são a chave primária composta.

{DateTime, Place} e TeamID são chaves estrangeiras.

TeamWithdrawal([DateTime, Place]→TeamParticipation, 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, CName, Country, BirthDate)

CID → CName, Country, BirthDate

Groom (GID, GName, BirthDate)

GID → GName, BirthDate

Sponsor(SID, Job, NIF)

SID → Job, NIF

NIF → SID

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

Vet(VID, VName, PhoneNumber, AppointmentCost)

VID → VName, PhoneNumber, AppointmentCost

PhoneNumber → VID

A segunda dependência funcional não viola a Forma Normal de Boyce-Codd já que existe uma chave do lado esquerdo da mesma. Analisando o closure de PhoneNumber

"{PhoneNumber}⁺ = {PhoneNumber, VID, VName, AppointmentCost}", estão incluídos todos os atributos da tabela e por isso não é violada a BCNF.

Club(CName, CreationYear)

CName → CreationYear

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

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

Team(TeamID, TName, AmountGain, TotalPoints)

TeamID → TName, AmountGain, TotalPoints

TName → TeamID

A segunda dependência funcional não viola a Forma Normal de Boyce-Codd já que existe uma chave do lado esquerdo da mesma. Analisando o closure de TName "{TName}⁺ = {TName, TeamID, AmountGain, TotalPoints}", estão incluídos todos os atributos da tabela e por isso não é violada a BCNF.

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

RiderID → RName, Country, NoFederated, BirthDate, AmountGain, TotalPoints, CoachID, VetID, GroomID, StableID, TeamID

NoFederated → RiderID, RName, Country, BirthDate, AmountGain, TotalPoints, CoachID, VetID, GroomID, StableID, TeamID

A segunda dependência funcional não viola a Forma Normal de Boyce-Codd já que existe uma chave do lado esquerdo da mesma. Analisando o closure de NoFederated "{NoFederated}⁺ = {NoFederated, RiderID, RName, Country, BirthDate, AmountGain, TotalPoints, CoachID, VetID, GroomID, StableID, TeamID}", estão incluídos todos os atributos da tabela e por isso não é violada a BCNF.

Sponsors(RiderID→Rider, SponsorID→Sponsor, Value)

RiderID, SponsorID → Value

Horse(HorseID, BirthDate, Breed, Gender, CoatColor, RiderID→Rider)

HorseID → BirthDate, Breed, Gender, CoatColor, RiderID

Participation(DateTime → Class, Place, Points, PTime)

$\text{DateTime, Place} \rightarrow \text{Points, PTime}$

$\text{DateTime, Points, PTime} \rightarrow \text{Place}$

A segunda dependência funcional não viola a Forma Normal de Boyce-Codd já que existe uma chave do lado esquerdo da mesma. Analisando o closure " $\{\text{DateTime, Points, PTime}\}^+ = \{\text{DateTime, Points, PTime, Place}\}$ ", estão incluídos todos os atributos da tabela e por isso é não violada a BCNF.

IndividualParticipation($[\text{DateTime, Place}]$ \rightarrow Participation, $\text{RiderID} \rightarrow \text{Rider}$,
 $\text{HorseID} \rightarrow \text{Horse}$)

$\text{DateTime, Place} \rightarrow \text{RiderID, HorseID}$

IndividualDisqualification($[\text{DateTime, Place}]$ \rightarrow IndividualParticipation, $\text{RiderID} \rightarrow \text{Rider}$)

$\text{DateTime, Place} \rightarrow \text{RiderID}$

IndividualWithdrawal($[\text{DateTime, Place}]$ \rightarrow IndividualParticipation, $\text{RiderID} \rightarrow \text{Rider}$)

$\text{DateTime, Place} \rightarrow \text{RiderID}$

TeamParticipation($[\text{DateTime, Place}]$ \rightarrow Participation, $\text{TeamID} \rightarrow \text{Team}$)

$\text{DateTime, Place} \rightarrow \text{TeamID}$

HorseParticipation($[\text{DateTime, Place}]$ \rightarrow TeamParticipation, HorseID \rightarrow Horse)

Sem dependências funcionais não triviais.

TeamDisqualification($[\text{DateTime, Place}]$ \rightarrow TeamParticipation, $\text{TeamID} \rightarrow \text{Team}$)

$\text{DateTime, Place} \rightarrow \text{TeamID}$

TeamWithdrawal($[\text{DateTime, Place}]$ \rightarrow TeamParticipation, $\text{TeamID} \rightarrow \text{Team}$)

$\text{DateTime, Place} \rightarrow \text{TeamID}$

Season(Year)

Sem dependências funcionais não triviais.

Event(City, Year \rightarrow Season, BeginDate, EndDate)

$\text{City, Year} \rightarrow \text{BeginDate, EndDate}$

$\text{BeginDate, EndDate} \rightarrow \text{Year, City}$

$\text{BeginDate} \rightarrow \text{Year, City}$

$\text{EndDate} \rightarrow \text{Year, City}$

A segunda, terceira e quarta dependências funcionais não violam a Forma Normal de Boyce-Codd, uma vez que existe sempre uma chave do lado esquerdo da mesma. Analisando os closures:

$\{\text{BeginDate, EndDate}\}^+ = \{\text{BeginDate, EndDate, Year, City}\}$

$\{\text{BeginDate}\}^+ = \{\text{BeginDate, EndDate, Year, City}\}$

$\{\text{EndDate}\}^+ = \{\text{BeginDate, EndDate, Year, City}\}$

Conclui-se que em todos os casos todos os atributos da tabela estão incluídos, pelo que não é violada a BCNF.

Class(DateTime, [City, Year] \rightarrow Event, MaxTime, NoObstacles)

$\text{DateTime} \rightarrow \text{City, Year, MaxTime, NoObstacles}$

Prize(Place, Class \rightarrow Class, Value)

$\text{Place, Class} \rightarrow \text{Value}$

$\text{Value, Class} \rightarrow \text{Place}$

A segunda dependência funcional não viola a Forma Normal de Boyce-Codd já que existe uma chave do lado esquerdo da mesma. Analisando o closure “ $\{\text{Value, Class}\}^+ = \{\text{Value, Class, Place}\}$ ”, estão incluídos todos os atributos da tabela e por isso não é violada a BCNF.

Pode-se afirmar que nenhuma relação apresentada anteriormente viola a 3ª Forma Normal já que todos os atributos não primos dependem de todas as chaves primárias de uma forma não transitiva ou, tendo $\bar{A} \rightarrow \bar{B}$, \bar{A} é uma chave ou \bar{B} é constituído apenas por atributos primos (membro de alguma 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)) e de ser único (restrição UNIQUE).

Vet

- VID é a chave primária (restrição PRIMARY KEY), isto é, não pode ser nula e tem de ser única;
- VName, PhoneNumber e AppointmentCost não podem ser nulos (restrição NOT NULL);
- AppointmentCost não pode ser negativo (restrição CHECK AppointmentCost >= 0);
- PhoneNumber tem de ser único (restrição UNIQUE) e tem de ter 9 dígitos (restrição CHECK (length(PhoneNumber) = 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 e NoEmployees não podem ser NULL (restrição NOT NULL) nem negativos (restrições: CHECK NoHorses >= 0, CHECK NoPractitioners >= 0 e CHECK NoEmployees >= 0);
- InaugurationYear e Area não pode ser nulo (restrição NOT NULL) e tem de ser maior do que 0 (restrições: CHECK InaugurationYear > 0 e CHECK Area > 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).

- VetID tem de ser único (restrição UNIQUE), uma vez que cada veterinário assiste apenas um cavaleiro.

Sponsors

- RiderID e SponsorID são uma chave primária composta (restrição PRIMARY KEY(RiderID, SponsorID)), pelo que nenhum destes atributos pode ser nulo; para além disso são também chaves estrangeiras (restrição FOREIGN KEY).
- Value tem de ser positivo (restrição CHECK Value > 0) e não nulo (restrição NOT NULL).

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);
- BirthDate, Breed e CoatColor não podem ser nulos (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);
- 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)), pelo que nenhum destes atributos pode ser nulo;
- 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)), pelo que nenhum destes atributos pode ser nulo; para além disso 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)), pelo que nenhum destes atributos pode ser nulo; para além disso 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)), pelo que nenhum destes atributos pode ser nulo; para além disso 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)), pelo que nenhum destes atributos pode ser nulo;
- 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)), pelo que nenhum destes atributos pode ser nulo; para além disso 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)), pelo que nenhum destes atributos pode ser nulo; para além disso 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)), pelo que nenhum destes atributos pode ser nulo; para além disso são também 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)), pelo que nenhum destes atributos pode ser nulo;
- 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), pelo que nenhum destes atributos pode ser nulo;
- 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).

Nas restrições do tipo foreign key, é também especificado como se irá comportar esta base de dados no caso de alteração ou remoção. No caso de alteração (*UPDATE*), é utilizado valor *CASCADE* já que, se houver alguma alteração na chave pai, esta deve ser propagada para os filhos que de si dependam. No caso de remoção (*DELETE*), é utilizado o valor default (*RESTRICT*) nos casos em que os atributos estão definidos como *NOT NULL*. Em certos casos, é utilizado o *CASCADE*, porque há relações onde não faz sentido existir uma parte se a outra parte for apagada. Por exemplo, no caso da relação entre Horse e Rider: se um cavaleiro for eliminado, não faz sentido guardar a informação acerca dos seus cavalos, uma vez que mais ninguém poderá competir com

eles. Ainda noutros casos, é utilizado o *SET NULL*, casos estes em que a informação eliminada não é imprescindível para a existência da relação.

Para além das chaves primárias que pela sua condição não podem ser nulas, outros atributos foram identificados como não podendo ser nulos já que são informações importantes para implementação desta base de dados.

Interrogação da Base de dados

As 10 interrogações implementadas estão descritas abaixo em linguagem natural. Muitas delas utilizam a agregação *count*, uma vez que grande parte das questões pertinentes para a base de dados são estatísticas.

Interrogação 1

Qual o melhor cavaleiro em cada etapa, competindo individualmente?

Interrogação 2

Quais os pares de cavaleiros que participaram na mesma prova e qual a sua frequência?

Interrogação 3

Quais os cavaleiros com os 3 ganhos mais altos, tendo em conta o total de pontos e dinheiro adquiridos?

Interrogação 4

Qual a percentagem de desistência por prova individual?

Interrogação 5

Quais os cavaleiros que participaram numa prova de equipas, mas nunca participaram numa prova individual?

Interrogação 6

Quais os cavaleiros que participaram em todas as etapas?

Interrogação 7

Qual o melhor cavaleiro de cada clube?

Interrogação 8

Ordenar os países por número de classificações.

Interrogação 9

Qual o top 3 de treinadores, isto é, treinadores com maior taxa de sucesso?

Interrogação 10

Qual o número de participações por raça?

Adição de gatilhos à base de dados

Foram implementados os seguintes gatilhos:

Gatilho 1

Quando um novo tuplo é inserido na tabela *IndividualParticipation*, este gatilho atualiza o total de pontos e a quantia arrecadada pelo cavaleiro a que esta *IndividualParticipation* corresponde, uma vez que estes atributos são derivados. Assim, são acrescentados a estes dois atributos os valores referentes à nova participação inserida.

Gatilho 2

Quando se tenta inserir um novo tuplo na tabela *IndividualWithdrawal*, este gatilho verifica se existe um tuplo associado na tabela *Participation* e, caso não exista, esta inserção é ignorada. Tal acontece, uma vez que para desistir ou ser desqualificado é necessário participar.

Gatilho 3

Quando se tenta inserir um novo tuplo na tabela *TeamParticipation*, este gatilho verifica se existe um tuplo nesta tabela com o *TeamID* igual ao do tuplo que se está a tentar inserir, se já existir esta inserção é ignorada, já que não é possível uma equipa participar duas vezes na mesma prova.

Para além destes gatilhos, deviam também ser implementados um gatilho semelhante ao primeiro para a inserção de um tuplo na tabela *TeamParticipation* e gatilhos semelhantes a estes para remoção de tuplos destas mesmas tabelas. Na mesma ideia, é necessário gatilhos semelhantes ao segundo para inserção de tuplos nas tabelas *IndividualDisqualification*, *TeamDisqualification* e *TeamWithdrawal*, bem como gatilhos semelhantes ao terceiro para as restantes classes (*IndividualParticipation*, *IndividualWithdrawal*, *IndividualDisqualification*, *TeamDisqualification*, *TeamWithdrawal*).