



Jogos Olímpicos 2020

Grupo 907:

Afonso Abreu up202008552@edu.fe.up.pt

André Ávila up202006767@edu.fe.up.pt

João Malva up202006605@edu.fe.up.pt

Professor:

Michel Ferreira mpferrei@fc.up.pt

Carla Lopes ctl@fe.up.pt

Índice

1. Contexto
2. Diagrama UML
3. Esquema Relacional
4. Dependências Funcionais e Formas Normais
5. Restrições

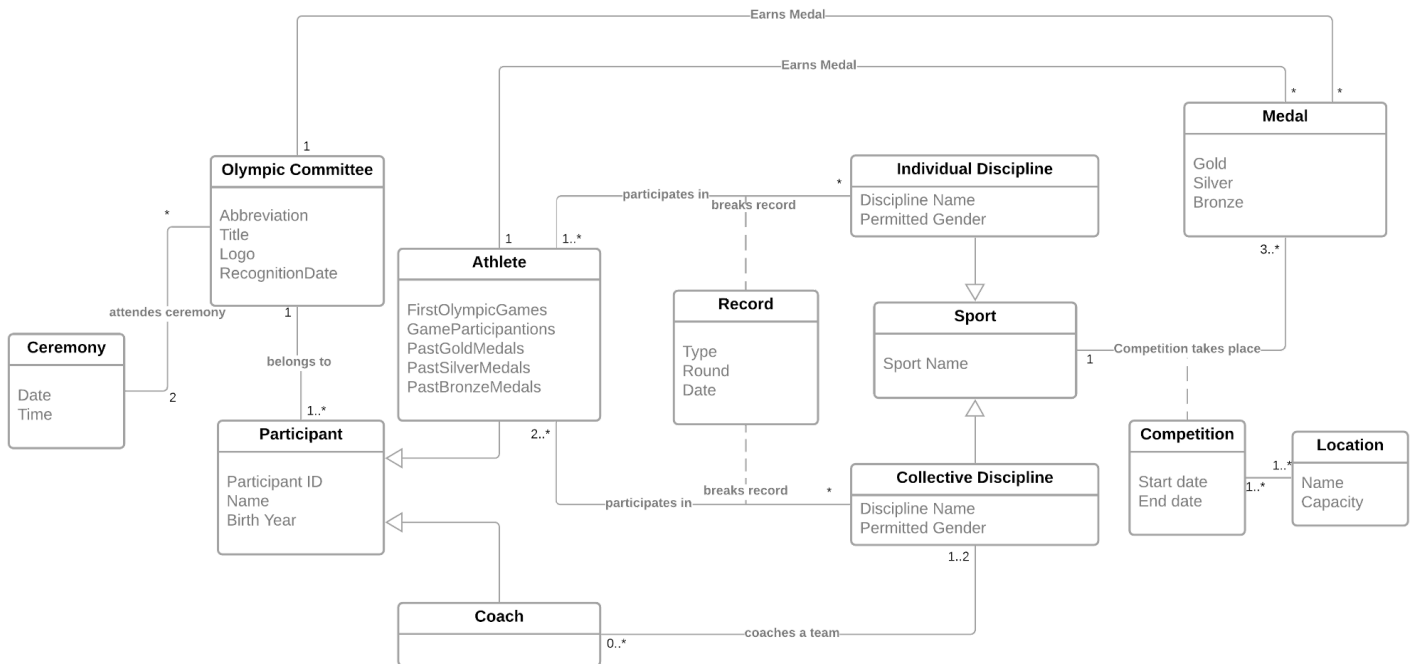
Contexto

O nosso projeto aborda o tema “Jogos Olímpicos 2020”, que decorreram em 2021, em Tóquio, no Japão. É um tema que consideramos apelativo devido ao imenso público que atrai em vários países independentemente da cultura que neles vigora. Também o facto dos Jogos Olímpicos serem um sistema de vencedores e perdedores é algo que combina com um modelo de base de dados e de registo dos mesmos, podendo incrementar o potencial deste projeto.

Este trabalho vai abordar os resultados finais das modalidades, sendo desprezados os eventos dos quais não se atribuíram medalhas, ou seja, estão apenas presentes na base dados as medalhas que um atleta ganhou e a modalidade.

A nossa base de dados deve armazenar dados sobre as várias modalidades (sports), os treinadores (coach), os atletas (athlete) o local onde decorreram os eventos (location), as cerimónias (ceremony) e os recordes batidos (records). Dos vários comités olímpicos queremos ter os atletas e os treinadores que os representam nas respetivas modalidades, para além de alguns dados únicos e de referência de cada um. Pretendemos guardar dados dos participantes das atividades Olímpicas, atletas e treinadores, como idade, género, medalhas obtidas em jogos anteriores entre outros (sendo este segundo importante devido à divisão de atletas pelo mesmo em certas categorias).

Diagrama UML



Esquema Relacional

Ceremony(idCeremony, date, time)

OlympicCommittee(abbreviation, title, logo, recognitionDate)

AttendesCeremony(abbreviation -> OlympicCommittee, idCeremony -> Ceremony)

Participant(idParticipant, abbreviation -> OlympicCommittee, name, birthYear)

Athlete(idParticipant -> Participant, firstOlympicGames, gameParticipations, pastGoldMedals, pastSilverMedals, pastBronzeMedals)

Coach(idParticipant -> Participant)

Sport(idSport, sportName)

Coaches(idParticipant -> Participant, idSport -> Sport)

ParticipatesIn(idParticipant -> Participant, idSport -> Sport)

Record(idParticipant -> Participant, idSport -> Sport, recordType, round, date)

IndividualDiscipline(idSport -> Sport, disciplineName, permittedGender)

CollectiveDiscipline(idSport -> Sport, disciplineName, permittedGender)

Competition(idSport -> Sport, startDate, endDate)

CompetitionLocation(idCompetition -> Competition, idLocation -> Location)

Location(idLocation, name, capacity)

Medal(idMedal, gold, silver, bronze)s

AthleteEarnsMedal(idMedal -> Medal, idParticipant -> Participant)

CommitteeEarnsMedal(idMedal -> Medal, abbreviation -> OlympicCommittee)

Dependências Funcionais e Formas Normais

Ceremony(idCeremony, date, time)

idCeremony -> {date, time} : Obedece à 3ª forma normal e à forma de Boyce-Codd.

OlympicCommittee(abbreviation, title, logo, recognitionDate)

abbreviation -> {title, logo, recognitionDate} : Obedece à 3ª forma normal.

title -> {abbreviation, logo, recognitionDate} : Obedece à 3ª forma normal.

logo -> {title, abbreviation, recognitionDate} : Obedece à 3ª forma normal.

Participant(idParticipant, abbreviation, name, birthYear)

idParticipant -> {abbreviation, name, birthYear} : Obedece à 3ª forma normal e à forma normal de Boyce-Codd.

Athlete(idParticipant, firstOlympicGames, gameParticipations, pastGoldMedals, pastSilverMedals, pastBronzeMedals)

idParticipant -> {firstOlympicGames, gameParticipations, pastGoldMedals, pastSilverMedals, pastBronzeMedals} : Obedece à 3ª forma normal de Boyce-Codd.

Record(idParticipant, idSport, recordType, round, date)

{idParticipant, idSport} -> {recordType, round, date} : Obedece à 3ª forma normal.

Sport(idSport, sportName)

idSport -> sportName : Obedece à 3ª forma normal e a forma normal de Boyce-Codd.

IndividualDiscipline(idSport, disciplineName, permittedGender)

idSport -> {disciplineName, permittedGender} : Obedece à 3ª forma normal e à forma normal de Boyce-Codd.

CollectiveDiscipline(idSport, sportName, disciplineName, permittedGender)

idSport -> {disciplineName, permittedGender} : Obedece à 3ª forma normal e à forma normal de Boyce-Codd.

Competition(idSport, startDate, endDate)

idSport -> {startDate, endDate} : Obedece à 3ª forma normal e à forma normal de Boyce-Codd.

Location(idLocation, name, capacity)

idLocation -> {name, capacity} : Obedece à 3ª forma normal e à forma normal de Boyce-Codd.

CompetitionLocation(idCompetition, idLocation)

idSport -> {disciplineName, permittedGender} : Obedece à 3ª forma normal e à forma normal de Boyce-Codd.

Medal(idMedal, gold, silver, bronze)

idMedal -> {gold, silver, bronze} : Obedece à 3ª forma normal e à forma normal de Boyce-Codd.

AthleteEarnsMedal(idMedal, idParticipant)

idMedal -> idParticipant : Obedece à 3ª forma normal e à forma normal de Boyce-Codd.

CommitteeEarnsMedal(idMedal, abbreviation)

idMedal -> abbreviation : Obedece à 3ª forma normal e à forma normal de Boyce-Codd.

Todas estas relações obedecem à 3ª forma normal porque os seus atributos têm um nome diferente e são atômicos (1ª forma normal) e não existem dependências parciais (2ª forma normal) nem dependências transitivas.

A relações que obedecem à forma de Boyce-Codd obedecem pois contêm uma superclasse (parte esquerda da dependência funcional) que não depende de nenhum outro atributo.

Restrições

Ceremony

- idCeremony: PRIMARY KEY -> Cada cerimónia é identificada por um valor único;
- date: NOT NULL -> São conhecidas todas as datas em que as cerimónias se deram;
- time: NOT NULL -> Todas as cerimónias ocorreram a uma determinada hora conhecida.

OlympicCommittee

- abbreviation: PRIMARY KEY -> Expressão derivada do título, diferente em cada comité;
- title: NOT NULL, UNIQUE -> Todos os comités têm um título que os distingue dos outros;
- logo: NOT NULL, UNIQUE -> Os comités são todos representados pelo seu próprio logotipo.

AttendsCeremony

- idCeremony: NOT NULL, FOREIGN KEY -> Relaciona a presente classe com a classe Ceremony;
- OlympicCommitteeAbbreviation: NOT NULL, FOREIGN KEY -> Relaciona a presente classe com a classe OlympicCommittee.

Participant

- idParticipant: PRIMARY KEY -> Número de ID único do participante;
- abbreviation: NOT NULL, FOREIGN KEY -> Todos os participantes representam o Comité Olímpico a qual pertencem;
- name: NOT NULL -> Todos os participantes possuem um nome.

Athlete

- idParticipant: PRIMARY KEY, FOREIGN KEY -> Todos os atletas são participantes e, por isso, estão relacionados com a classe Participant;
- pastGOLDmedals: NOT NULL -> O atleta possui sempre 0 ou mais medalhas de ouro;
- pastSILVERmedals: NOT NULL -> O atleta possui sempre 0 ou mais medalhas de prata;
- pastBRONZEmedals: NOT NULL -> O atleta possui sempre 0 ou mais medalhas de bronze.

Coach

- idParticipant INT PRIMARY KEY, FOREIGN KEY -> Todos os treinadores também são membros da classe Participant.

Sport

- idSport: PRIMARY KEY -> Cada desporto é identificado por um valor único;
- sportName: NOT NULL -> Cada desporto tem um nome que o representa.

Coaches

- idParticipant: NOT NULL, FOREIGN KEY -> Um participante treina pelo menos num desporto;
- idSport: NOT NULL, FOREIGN KEY -> Num desporto existe pelo menos um treinador.

ParticipatesIn

- idParticipant: NOT NULL, FOREIGN KEY -> Cada atleta participa em pelo menos um desporto;
- idSport: NOT NULL, FOREIGN KEY -> Num desporto compete pelo menos um atleta.

Record

- idParticipant: NOT NULL, FOREIGN KEY -> É sempre um atleta que bate um determinado recorde;
- idSport: NOT NULL, FOREIGN KEY -> Um recorde é associado a um desporto;
- recordType: NOT NULL -> Existe sempre um tipo de recorde batido (ex: recorde mundial, olímpico, etc.);
- round: NOT NULL -> O recorde é associado a uma ronda de uma certa modalidade;
- date: NOT NULL -> Um recorde é batido num determinado dia conhecido.

IndividualDiscipline

- idSport: NOT NULL, FOREIGN KEY -> Uma disciplina individual faz parte de um desporto e portanto, pertence à superclasse Sport;
- disciplineName: NOT NULL -> Cada modalidade tem um nome específico.

CollectiveDiscipline

- idSport: NOT NULL, FOREIGN KEY -> Uma modalidade coletiva faz parte de um desporto e portanto, pertence à superclasse Sport;
- disciplineName: NOT NULL -> Cada modalidade tem um nome específico.

Competition

- idSport: PRIMARY KEY ,FOREIGN KEY -> Cada competição está associada a um desporto. Pertence à classe Sport.

Location

- idLocation: PRIMARY KEY -> Uma localização é identificada por um valor único;
- name: NOT NULL -> Um certo nome está associado com uma localização.

CompetitionLocation

- idCompetition: NOT NULL, FOREIGN KEY -> Número não nulo que indica a competição respectiva. Pertence à classe Competition;
- idLocation: NOT NULL, FOREIGN KEY -> Uma competição é realizada num certo local. Pertence à classe Location.

Medal

- idMedal: PRIMARY KEY -> Cada medalha tem um valor único;
- gold, silver, bronze: CHECK -> Apenas vai ser atribuído o número '1' a um tipo de medalha (gold ou silver ou bronze). Uma medalha não pode ser de dois tipos nem ter 0 tipos associados.

AthleteEarnsMedal

- idMedal: NOT NULL, FOREIGN KEY -> Medalha única associada ao atleta. Pertence à classe Medal;
- idParticipant: NOT NULL, FOREIGN KEY -> Um atleta pode ganhar uma ou mais medalhas (ou nenhuma). Pertence à superclasse Participant.

CommitteeEarnsMedal

- idMedal: NOT NULL, FOREIGN KEY -> Medalha única associada ao comité olímpico. Pertence à classe Medal;
- abbreviation: NOT NULL, FOREIGN KEY -> Um comité olímpico pode ganhar uma ou mais medalhas (ou nenhuma). Pertence à classe OlympicCommittee.