KU LEUVEN

FACULTY OF ECONOMICS AND BUSINESS

FACULTY OF ECONOMICS AND BUSINESS

Principles of Database Management Assignment (D0l62a)

MASTER OF INFORMATION MANAGEMENT

Pratheek Praveen KUMAR

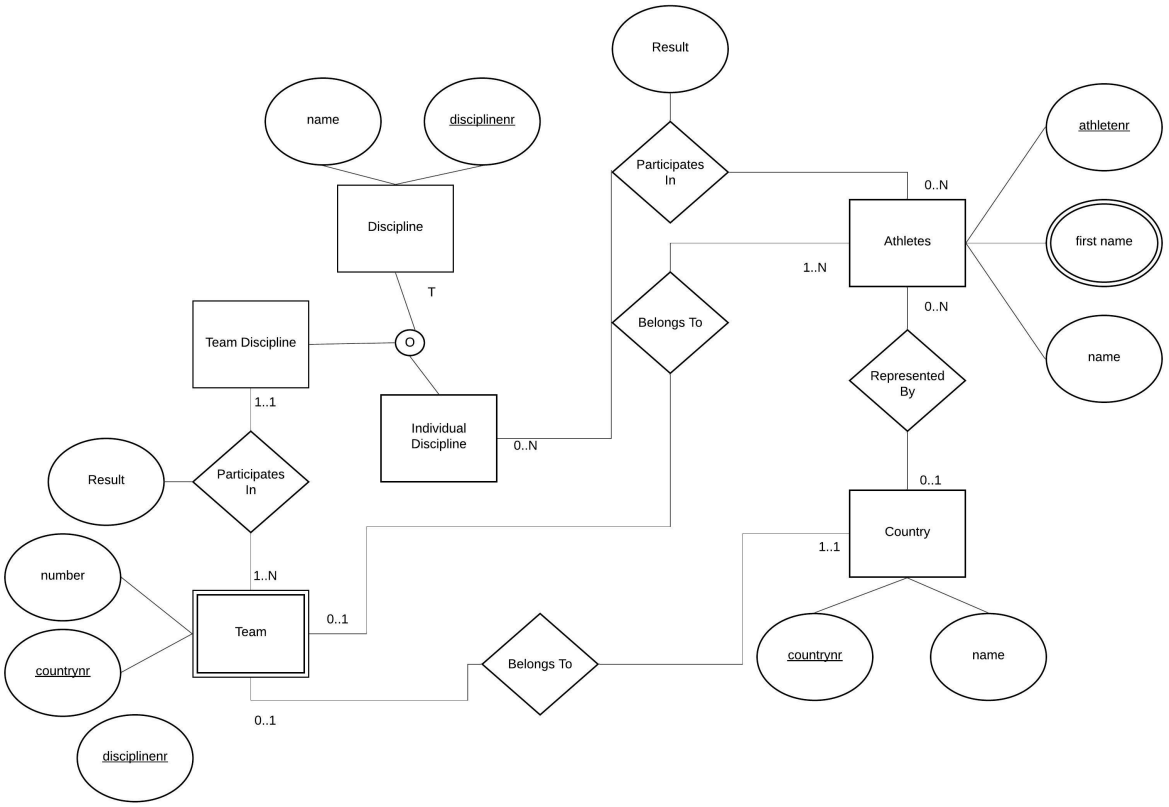
Dr. Matthias Bogaert

Academic year 2019-2020

Table of Contents

Question 1 (10/20)

1. Construct an EER model for the given problem. Document your solution. Extensively comment on semantics that cannot be enforced by the EER model, and illustrate with two examples.



In the above diagram, Disciplines, Athletes and Countries have been represented by entities. Team Disciplines and Individual Disciplines are the two subclasses specialised under the Discipline entity. The Athletes entity has athletenr, first name and name as its attributes, with athletenr being a key attribute and first name being a derived attribute. The country entity has two attributes, countrynr and name, with countrynr as its key attribute. The Discipline entity has two attributes, name and disciplinenr with disciplinenr being its key attribute. The result for both teams and individuals has been represented by an attribute of two "Participates in" relationships, one related to the Team Discipline and the other related to the Individual Discipline. Team has been modelled by using a weak entity type as it depends on the countrynr and disciplinenr attributes for its existence.

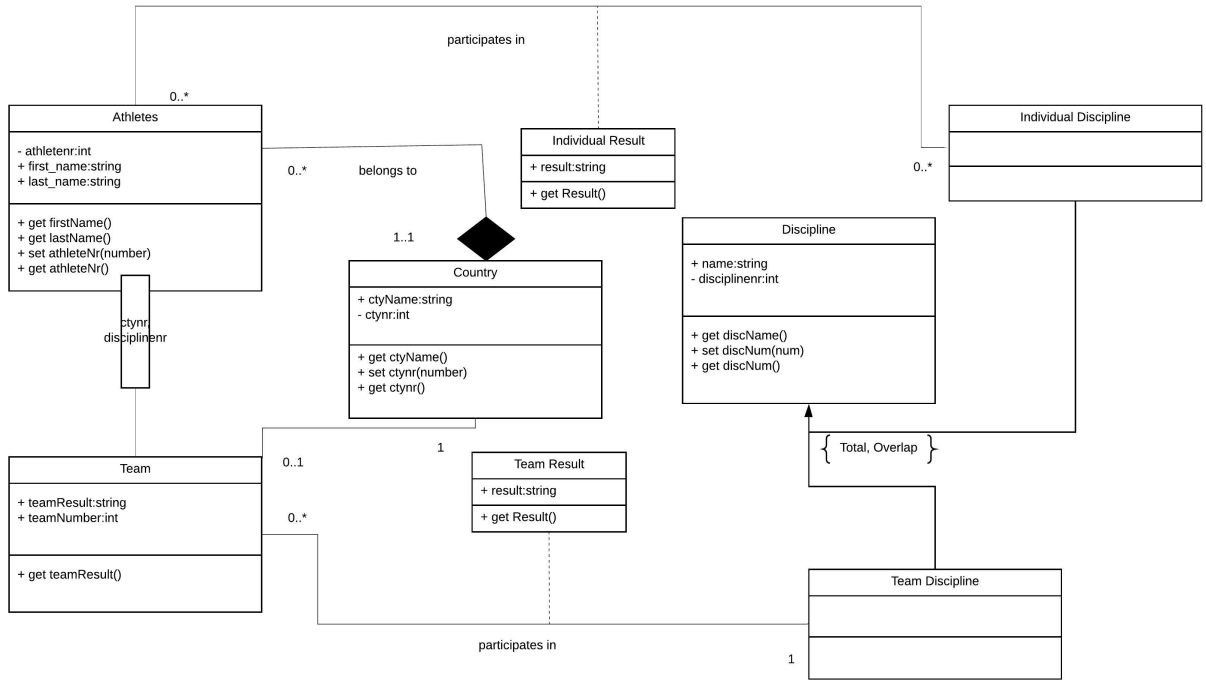
The EER model cannot model temporal constraints and also cannot guarantee the consistency across multiple relationship types. Also Domains and Functions are not included in the EER model.

For example,

1) In the above diagram, if it is required to limit the participation of certain countries to certain events only, we will not be able to ensure it. This shows loss of semantics.

2) Also, in the diagram, we cannot model the limit on the maximum number of athletes in each team across disciplines.

2. Construct an UML model for the given problem. Document your solution. Comment on semantics that cannot be enforced by the UML.



In the above diagram, Athletes, Countries, Disciplines, Team Disciplines, Individual Disciplines, Teams, Individual Results and Team Results have been modeled by using classes. Team Disciplines and Individual Disciplines are specializations of the Discipline class and are modeled as such. Individual Results and Team Results have been modeled as classes of associations. The relationship between the country and the athletes has been modeled using a composite aggregation as the athletes can only ever belong to one country. In the Athletes class, the first name and the last name have been modeled with public access modifiers so that everyone can use it. But the athlete number has been defined using a private access modifier so that no one can check the value. The same goes to the country number variable of the country class and the discipline number variable of the discipline class. This is to enforce information hiding. The other variables of the different classes have been made as public variables so that they can be accessed by everyone. The methods have all be modeled using the public access modifier so that they can all be accessed by everyone.

The UML specification is semantically richer than its EER counterpart. The weak entity type Team has been implemented using qualified associations and have the country number and discipline number as the key attribute. All associations have been defined as bidirectional which implies that they can be navigated in both directions. The UML class diagram can be further enriched with OCL constraints to implement the constraints that cannot be enforced by the EER model. We can specify invariants for classes, pre- and post-conditions for methods, to navigate between classes, or to define constraints on operations.

Question 2 (6/20)

Map the model to a relational model representation. Discuss the possible loss of semantics, or semantics that can be added. Clearly indicate the primary-foreign key relationships and specify NOT NULL declarations where necessary

CITY (Postal Code, Name, Mayor, Population, Province)

LOCAL_AUTHORITY_SERVICE (Name, Number Employees, Responsible, *Postal Code*)

Postal_Code refers to Postal_Code in CITY, NOT NULL

LEGAL ENTITY (ID, *Postal Code*, Address)

Postal_Code refers to Postal_Code in CITY, NOT NULL

INDIVIDUAL LEGAL ENTITY (*I_ID*, FirstName, LastName, DoB)

I_ID refers to ID in LEGAL ENTITY, NOT NULL

BUSINESS LEGAL ENTITY (*B_ID*, Name, Date of Registration, Business ID)

B_ID refers to ID in LEGAL ENTITY, NOT NULL

BUILDING PERMIT (ID, Zone, Address, *L_ID*)

L_ID refers to ID in LEGAL ENTITY, NOT NULL

COMMERCIAL BUILDING PERMIT (*B_ID*, *P_ID*, Sector, Safety policy, Parking, License Number)

B_ID refers to ID in LEGAL ENTITY, NOT NULL

P_ID refers to ID in BUILDING PERMIT, NOT NULL

ISSUES(*Name*, *Postal Code*, *P_ID*)

Name refers to Name in LOCAL AUTHORITY SERVICE, NOT NULL

P_ID refers to ID in BUILDING PERMIT, NOT NULL

Postal_Code refers to Postal_Code in CITY, NOT NULL

The following semantics were lost in the mapping:

1) the disjointness of the specialization of Legal Entity into Individual and Business cannot be enforced;

2) the disjointness of the specialization of Building Permit into Commercial Business Permit cannot be enforced;

3) the minimum cardinality of 1 in the relationship type Issues between Local Authority Service and Building Permit cannot be enforced;

4) cannot enforce that a city should have at least one legal entity.

5) cannot enforce that a city should have at least one local authority service.

Question 3 (4/20)

Give two examples of SQL queries as listed below, based on the EER model above. For each query, give the explanation of what the query does in plain English as well as in SQL.

1. One join query with GROUP BY AND HAVING

```
SELECT A.ArtistName, A.URL
FROM SONG P, ARTIST A
WHERE P.ArtistName = A.ArtistName
GROUP BY A.ArtistName

HAVING COUNT(*) > 5
```

This query selects the url and the name of the artist after doing an inner join on the artist and song tables and grouping the tuples by the artist name. All the artists who have released more than 5 songs are thus selected.

2. One correlated nested query

```
SELECT P. Title
FROM SONG P
WHERE Length >

(SELECT AVG(P2.Length)
FROM SONG P2
WHERE P.Year > 1995)
```

This query first finds out the average length of all the songs released from 1995. It then selects the title of the songs which have a length higher than it.