DBClass UML Quiz Answers

Andrew Grammenos
andreas.grammenos@gmail.com

Introduction

In this document I'll provide the solutions for the Universal Model Language (UML) quiz questions from the (infamous) dbclass MOOC kindly provided by Prof. Jennifer Widom.

Since markdown does not convey maths as it should, I created this document in order to preserve the intended notation as well as maintain cohesion; thus, since I got into all this trouble anyway solutions are a bit more detailed than the other (plain markdown) solutions in most chapters.

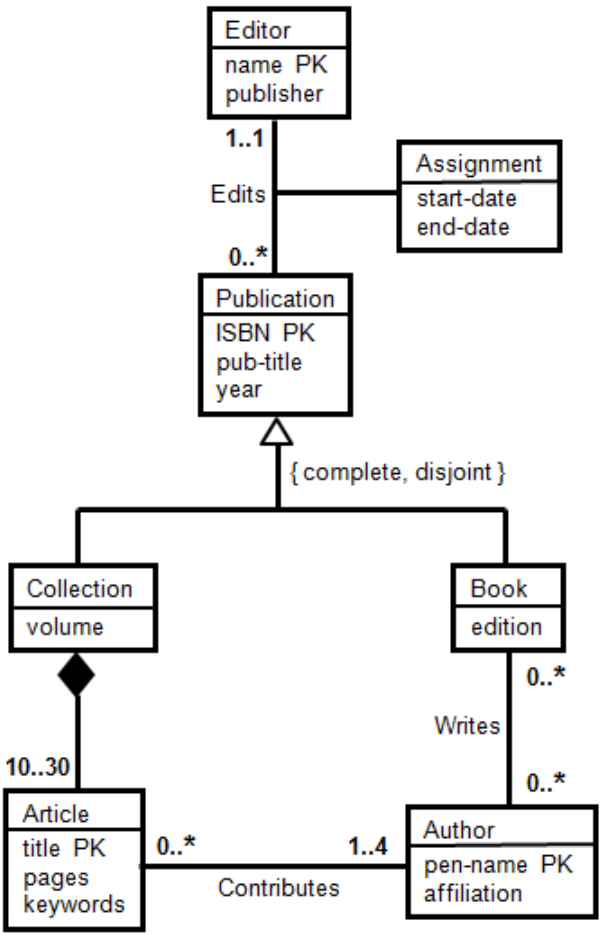
An **important** note is that the provided answers are the ones that were generated in my instance but the system is smart enough to generate *different* possible answers and mixes up the order as well, so your mileage may vary.

Finally you should use this documents (and my solutions for the dbcourse in general) as a *reference* and just copy-paste; you're just hurting yourself if you do so. Now without further delays, off we go!

Question 1

Consider translating the UML diagram shown into Figure 1 in relations. Which of the following relations would not be generated by any of the recommended translation schemes discussed in the lectures?

Figure 1



Q1 Options

In my instance the options to select the answer from, were the following:

a) $\mathcal{R}_a$: Book(ISBN, edition)

b) $\mathcal{R}_b$: Book(edition)

c) $\mathcal{R}_c$: Publication(ISBN, pub-title, year, name, start-date, end-state)

d) $\mathcal{R}_d$: Collection(ISBN, volume)

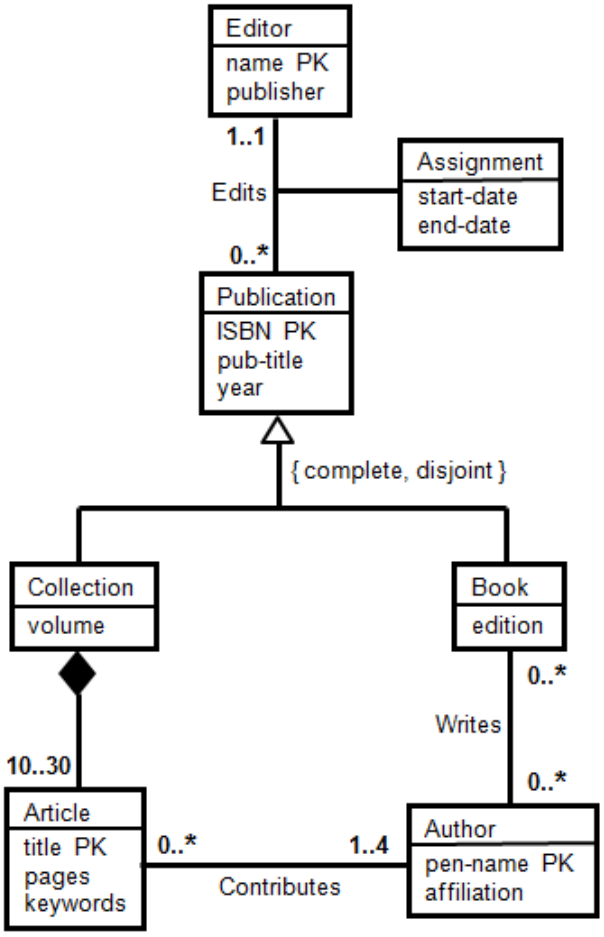
Q1 Answer

Let's examine the given answers one by one... first of both $\mathcal{R}_a$ and $\mathcal{R}_d$ are valid because if we elect to indeed create separate classes for `Book` and `Collection` we will borrow the key (ISBN) from parent class which is `Publication`. Additionally $\mathcal{R}_c$ is also valid as due to the 1..1 cardinality that `Publication` has to `Editor` we can glue the `Edits` association to it. The only invalid (and very obvious) violating option is $\mathcal{R}_b$, as `Book` is a subclass of `Publication`, hence when creating the relation it *must* have the primary key of the parent relation with it. Thus the correct answer is option: **b**, $\mathcal{R}_b$: Book(edition).

Question 2

Based on the UML that is shown into Figure 2, which of the following statements about `Authors` is correct?

Figure 2



Q2 Options

In my instance the options to select the answer from, were the following:

a) $\mathcal{S}_a$: An article may have any number of authors.

b) $\mathcal{S}_b$: An author can write a book or contribute an article, but not both.

c) $\mathcal{S}_c$: A book may have no authors.

d) $\mathcal{S}_d$: Every author has contributed up to four articles.

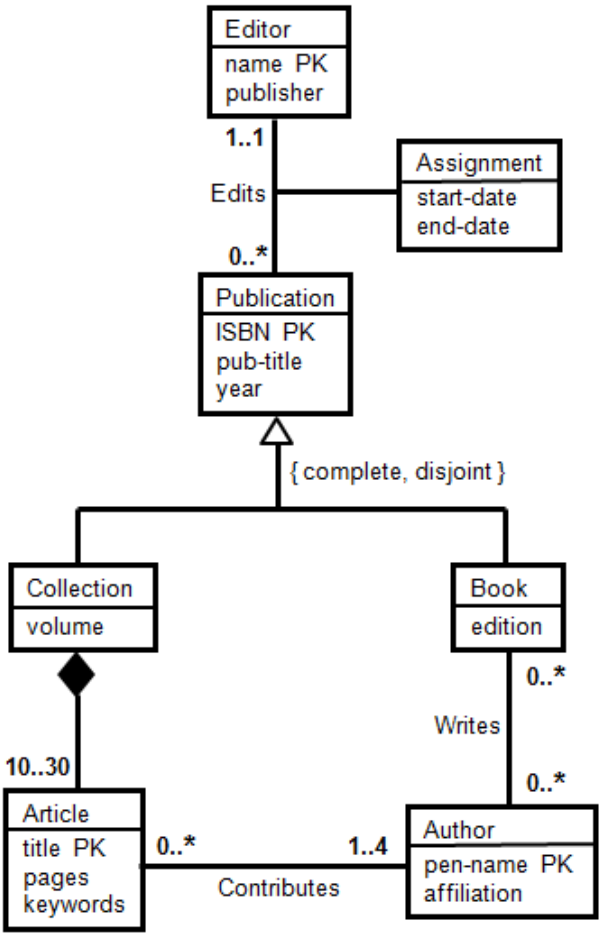
Q2 Answer

Let's examine the given answers one by one... The first option is not correct has based on the cardinalities we see that an `Article` can have 1..4 authors. Additionally the second option is again not correct as the `Author` has a cardinality of 0..$*$ against both `Article` and `Book` which essentially means that there can be zero or any number of authors for each article or book. Now moving onto the third option we can easily see that it holds as it's the same cardinality we used to invalidate the previous option. Finally the fourth and last option is again incorrect as `Author` has a cardinality of 0..$*$ and that means that an author can contribute to zero or any number of articles. Thus, the correct answer is option: **c**, $\mathcal{S}_c$: A book may have no authors.

Question 3

Based on the UML that is shown into Figure 3, which of the following relations best represents `Articles`?

Figure 3



Q3 Options

In my instance the options to select the answer from, were the following:

a) $\mathcal{R}_a$: Article(title, pages, keywords, volume, ISBN, pub-title, year)

b) $\mathcal{R}_b$: Article(title, pages, keywords, ISBN)

c) $\mathcal{R}_c$: Article(title, pages, keywords)

d) $\mathcal{R}_d$: Article(title, pages, keywords, volume)

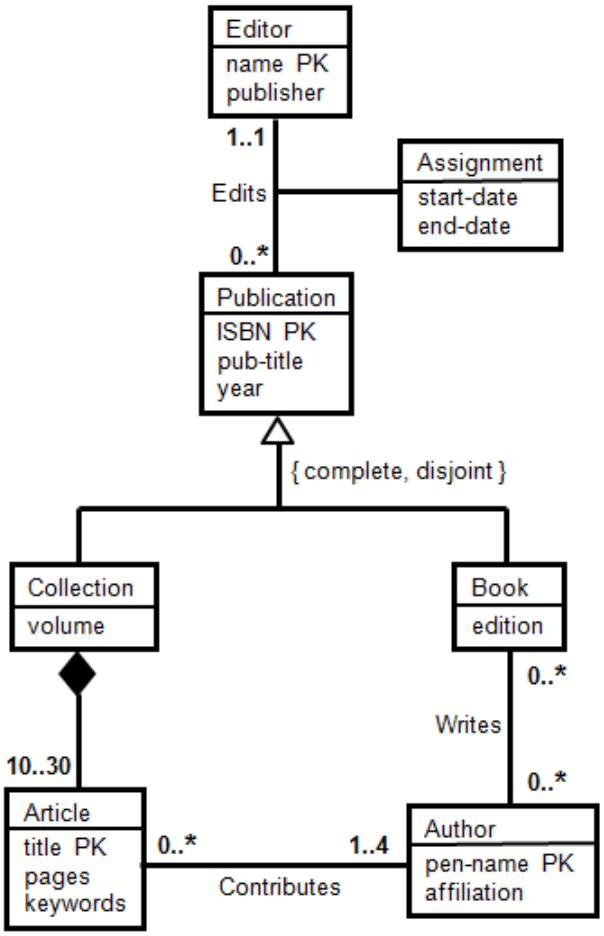
Q3 Answer

This is an easy one, if you recall the lectures in UML `Composition`-type relationships should contain its attributes plus the primary key of its including class. We can see from the diagram that the including class is `Collection`, which in turn inherits its primary key, ISBN, from `Publication` class. The only option that satisfies these constraints is option: **b**, $\mathcal{R}_b$: Article(title, pages, keywords, ISBN); which is also the correct answer.

Question 4

Consider translating the UML diagram shown into Figure 4 in relations. In the relation `Edits` generated from *Edits* association which of the following set of underlined attributes is a minimal key?

Figure 4



Q4 Options

In my instance the options to select the answer from, were the following:

a) $\mathcal{R}_a$: Edits(<u>name</u>, ISBN, <u>start-date</u>, <u>end-date</u>)

b) $\mathcal{R}_b$: Edits(name, <u>ISBN</u>, start-date, end-date)

c) $\mathcal{R}_c$: Edits(name, <u>ISBN</u>, <u>start-date</u>, <u>end-date</u>)

d) $\mathcal{R}_d$: Edits(<u>name</u>, <u>ISBN</u>, start-date, end-date)

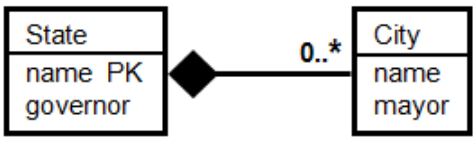
Q4 Answer

Again this answer comes straight from the rules indicated in our lectures. The default key in the relation when we have an association is the combination of the primary key of the classes this association applies to; remember though that if we have a $0..*$ cardinality in one end then we can just use as a key the one that's in that side and not both. Hence based on the what was said previously the only relation that satisfies our constraints is the one in option: **b**, $\mathcal{R}_b$: Edits(name, <u>ISBN</u>, start-date, end-date). Which is also the correct answer.

Question 5

Based on the UML diagram shown in Figure 5, which of the following statements about the `City` and `State` classes is correct?

Figure 5



Q5 Options

In my instance the options to select the answer from, were the following:

a) $\mathcal{S}_a$: No two cities can have the same mayor.

b) $\mathcal{S}_b$: Each state has at least one city.

c) $\mathcal{S}_c$: No person can be the mayor of cities in two different states.

d) $\mathcal{S}_d$: No two states can have the same name.

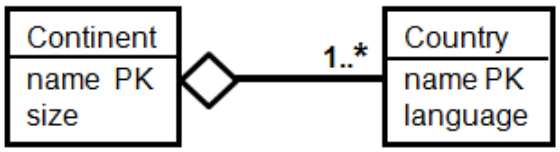
Q5 Answer

Let's first establish the cardinality of each side. We know from the lectures that the Composition side (the one with the hollow diamond) has a 1..1 cardinality; that basically means that a city can be part of *only one* state. Additionally, on the other side we have 0..∗, which means that `State` can have zero or any number of cities. Based on the previously mentioned constraints the only option that makes sense is option: **d**, $\mathcal{S}_d$: No two states can have the same name.; which is also the correct answer. Since the *name* attribute is the primary key of the class `State` it has to be unique by definition.

Question 6

Based on the UML diagram shown in Figure 6, which of the following statements about the `Continent` and `Country` classes is correct?

Figure 6



Q6 Options

In my instance the options to select the answer from, were the following:

a) $\mathcal{S}_a$: A country can speak two languages.

b) $\mathcal{S}_b$: Each country must belong to a continent.

c) $\mathcal{S}_c$: A country and a continent may have the same name.

d) $\mathcal{S}_d$: No two continents can have the same size.

Q6 Answer

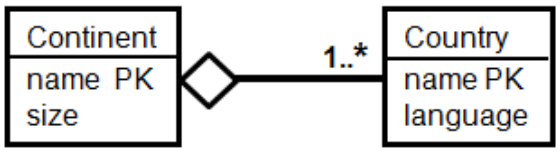
Let's first establish the cardinality of each side. We know from the lectures that the Aggregation side (the one with the hollow diamond) has a 0..1 cardinality; that basically means that a country can be part of zero or any continents. Additionally, on the other side we have 1..∗, which means that `Continent` can have *at least* one or any number of countries. Based on the previously mentioned constraints the only option that makes sense is option: **c**, $\mathcal{S}_c$: A country and a continent may have the same name; which is also the correct answer. Now, please don't be confused due to the fact that both relations use the same name for their primary key – this will be fixed during the translation (i.e. use for `Continent`: cont_name and for `Country`: c_name or similar).

Question 7

The UML diagram shown Figure 7 in puts some constraints on the cardinalities of classes `A`, `B`, and `C`. Which of the following combinations of cardinalities is permitted?

Note: The cardinality of a class `C`, denoted `|C|`, indicates the number of objects in the class.

Figure 7



Q7 Options

In my instance the options to select the answer from, were the following:

- $\mathcal{C}_a$: `|A|` $= 0$; `|B|` $= 20$; `|C|` $= 10$
- $\mathcal{C}_b$: `|A|` $= 0$; `|B|` $= 10$; `|C|` $= 1$
- $\mathcal{C}_c$: `|A|` $= 1$; `|B|` $= 0$; `|C|` $= 1$
- $\mathcal{C}_d$: `|A|` $= 0$; `|B|` $= 0$; `|C|` $= 10$

Q7 Answer

This is an easy answer, as the cardinalities bind the number of elements each class can have. Based on Figure 7 we can see that `A` has a 1..1 relationship with `B` while `B` has a $0..*$ relationship with `A`; hence this tells us that the number of objects in `A` **must** be less or equal than the objects contained in `B`.

In a similar fashion we can see that `B` has a 1..1 relationship with `C` while `C` has a 0..1 relationship with `B`. This tells is that the number of objects in `B` must be less or equal to the number of objects in `C`. Concretely, based on what we said the following inequality must hold:

$$\texttt{|A|} \leq \texttt{|B|} \leq \texttt{|C|} \tag{1}$$

Thus the only one of the given options that satisfies (1), is option: **d**, $\mathcal{C}_d$: `|A|` $= 0$; `|B|` $= 0$; `|C|` $= 10$; which is the correct answer.

Question 8

Suppose there is a UML superclass `Movies` with subclasses. Consider the following possible pairs of subclasses:

1. {B,NB}: B = movies in which Kevin Bacon appears; NB = movies in which Kevin Bacon does not appear
2. {B,R}: B = movies in which Kevin Bacon appears; R = movies in which Julia Roberts appears
3. {B,K}: B = movies in which Kevin Bacon appears; K = movies in which Val Kilmer appears
4. {L,S}: L = movies more than 100 minutes long; S = movies less than 105 minutes long

Consider whether each pair of subclasses is *complete* or *incomplete* (partial), and whether the pair is *overlapping* or *disjoint* (exclusive). Depending on your knowledge, you may have to do some web searches on movies to get the right classification. Which of the following statements is correct?

Q8 Options

In my instance the options to select the answer from, were the following:

- $\mathcal{P}_a$: {L, S} is incomplete and overlapping.
- $\mathcal{P}_b$: {B, NB} is complete and overlapping.
- $\mathcal{P}_c$: {B, R} is incomplete and disjoint.
- $\mathcal{P}_d$: {B, NB} is complete and disjoint.

Q8 Answer

This is a funny question; although, let's recall some notation first.

- *complete*: means that **all** objects are covered by the set properties
- *incomplete*: means that **some** objects are covered by the set properties
- *disjoint*: means that objects are in **only** one set class.
- *overlapping*: means that objects are in **one, or more** set class(es).

Now, logically the only correct definition based on the above notations is option: **d**, $\mathcal{P}_d$: {B, NB} is complete and disjoint. That's because Kevin Bacon can be in a movie cast or it cannot – so the set covers **all** objects (movies) and thus is *complete* while since it has a binary relation – can be in a movie cast or can't it is also *disjoint*. I'll leave you to find why the other options don't hold :).