Data & Information – Test 2 (1.5 hours)

17 May 2019, 13:45–15:15

Program: Technical Computer Science / Business & IT
Module: Data & Information (201700279)
Module Coordinator: Klaas Sikkel

Please note:

- Please answer every question on a different sheet of paper (the answers will be distributed to different person for grading).
- You are not allowed to bring any study materials to the test; essential excerpts from the study materials are available as appendices. You do not need a calculator.

Grade = #points/10

Introduction: University admission for international students

Question 1, 2, and 3a make use of the following case description.

For students from the Netherlands, there are standard procedures to enroll in a university program. If you come from another country it is more complicated – an *admission board* has to decide whether applicants fulfil the prerequisites, based on the evidence that the applicants can provide of their previous education.
The number of international students has increased rapidly over the last few years, and a new system is needed to support the handling of these applications. You are asked to make an initial design based on the following information

Any person who wants to apply for a study program at the University can file an application through the website. Some personal information has to be entered into a web form, copies of documents that support the application can be provided as PDFs.
Decisions on submitted applications are taken by the *admission board* for the requested program. Admission boards consist of staff members, typically educational directors and teachers, qualified to make decisions.
An admission board is aided by an admission coordinator, who does most of the work in the admission process, but is not member of the admission board. The admission coordinator prepares the meetings of the admission board and makes sure that the board members have the right information beforehand.
When a decision has been taken by the admission board, the admission coordinator writes (the contents of) a letter explaining the decision to the applicant. The letter will be dispatched by the central student administration.

Question 1 (Database Schema) (30 points)

A partial class diagram for the administation of applications is shown in Figure 1.
Applicants are asked for some personal details, for the name of the school and the country where they did their secondary education, as wel as the type of diploma. Also it is stored when the application was submitted and for which program. If it is an application for a master program, the bachelor degree and the university where this was obtained need to be filled in as well.
When a decision about the application has been made, the following information is stored: the date; the decision; the text of a letter in which the applicant is informed about the decision.
The University offers other kinds of programs besides master programs (definitely bachelor programs, possibly post-graduate programs, perhaps others), but for these only the stated attributes of *Application* are required, no additional information. the different kinds of programs are disjoint, i.e., a program can never be both a bachelor and a master program.

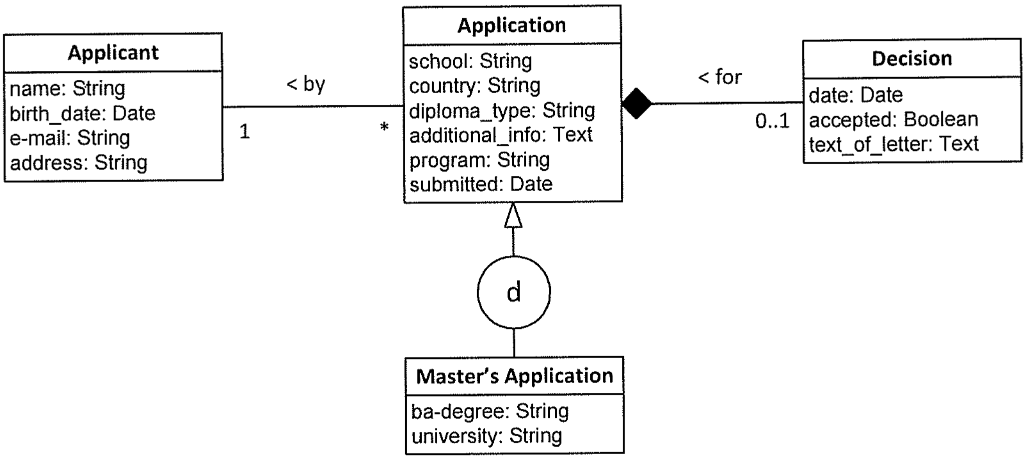


Figure 1: Partial class diagram for the administration of university admissions

a) Define a database schema for the (partial) class diagram in Figure 1.

b) What alternative schemas would be possible for the database schema that you gave under a) ? For each possible alternative,

- Describe in half a line what the alternative would be,
- Describe what would change in your database schema

Question 2 (Class Diagram) (35 points)

Complete the class diagram, given the information below.
Please copy the classes from Figure 1 into you class diagram.
Hints:

- *Be aware that some attributes in the given classes may change in the light of the new information provided*
- *If different classes share a substantial number of attributes, please model this by means of a generalization.*
- *Please pay attention to the difference between multiplicities * and 1..**

An application for admission for a study program is for one particular year. For example: if you are admitted to the master program BIT for 2019/20, you can start 1 Sep 2019 or 1 Feb 2020. The rules for the 2019/20 program will remain the same for all students who started that year. In other words, there could be changes in the program in 2020/21, but these changes affect only students who will be admitted to the master program BIT 2020/21 and do not affect students who were already admitted in 2019/20.

Due to various problems it could happen that an admission gets delayed, and the same person successfully applies for admission for the next year. This is administered as two separate applications.

In addition to what is already mentioned above and what is specified in Figure 1, the following information is relevant for the class diagram.

- For each application a number of separate documents have to be uploaded: copies of diplomas, passport, etc. Each such document has a document name and a document content (PDF). Each document belongs to one specific application (if the same document would be needed for two applications, two separate copies have to be uploaded).
- Study programs are identified by a (three-letter) abbreviation and also have full name. For each program is it known whether it is a bachelor or a master program. For each academic year in which the program is offered, one or two start dates are given (usually this is 1 September. In some cases it is also possible to start 1 February. It could happen, though, that a program offers an additional start in February in some years, but not all years.)
- Each program has an admission coordinator, but this could be the same person for different programs (e.g. joint admission coordinator for TCS and BIT). An admission coordinator is a staff member of the University.
- Information about admission boards and their members is stored as well. An admission board is responsible for one or more programs (e.g. one admission board for the master programs Computer Science and Internet Science & Technology; one admission board for the bachelor and master programs Business & IT).

  Each admission board has at least three, possibly more members. Member of the admission board are staff members of the University. For each member it is known which qualification entitles him/her to be member of the admission board. (It could be that the same person is a member of different admission boards with different qualifications, e.g. a Ph.D in Computer Science as a qualification for the admission board Computer Science, industrial experience in IT as a qualification for the admission board BIT).
- For a staff member of the university the following information is stored: name, e-mail address, birth date, employee number, office number, and one or more telephone numbers. For each known telephone number, a description is stored (e.g. office, home, mobile, or whatever other description is needed).

  In principle it is possible that a staff member would apply for a study program.

  The database holds information only about staff members and applicants, not about other persons.

Question 3 (35 points)

3a) (Functional dependencies) (15 points)

For the administration of application to the University the relation *R*(*Apt*,*Apn*,*Doc*,*Prog*,*Coord*,*Board*,*Member*) is defined, with the following attributes:

Apt:	Applicant	*Prog*:	Program
Apn:	Application	*Coord*:	Admission coordinator
Doc:	(copy of) Document supplied with an application	*Board*:	Admission board
		Member:	Admission board member

Furthermore, the following facts are given:

1. An application is submited by one applicant
2. A document is linked to one application (i.e. if someone would submit two applications with the same supporting document, these have to be separate copies of the document)
3. An application is always for one particular program
4. A program has one admission board
5. A program has one admission coordinator

For each of the following – potential – functional dependencies s a)–h) and multivalued dependencies i)–j), indicate whether they hold ("yes") or not ("no"). Please motivate your answer, if possible referring to the statements 1–5 above.

a) $Doc \longrightarrow Apt$

b) $Apt \longrightarrow Doc$

c) $Coord \longrightarrow Prog$

d) $Apt\ Prog \longrightarrow Coord$

e) $Doc \longrightarrow Coord$

f) $Member\ Board \longrightarrow Prog$

g) $Board\ Prog \longrightarrow Member$

h) $Prog\ Member \longrightarrow Board$

i) $Apn \twoheadrightarrow Apt\ Doc\ Proc$

j) $Apn \twoheadrightarrow Member\ Board$

3b) (Normal forms) (20 points)

Consider the relational schema *R*(*A*,*B*,*C*,*D*) with functional dependencies $\mathfrak{F}$, defined by

$$\mathfrak{F} = \{\, A \rightarrow B,\ B \rightarrow C,\ BD \rightarrow A \,\}.$$

1) Compute $\mathfrak{F}^{+}$ and determine the candidate keys.
Which functional dependencies violate the BCNF condition?

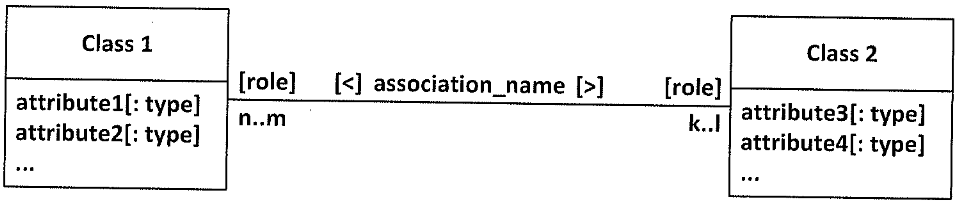
2) Apply the algorithm in appendix C (p. 6) to decompose *R* into a set of relational schemas that are all in BCNF. For each decomposition step, please give the resulting schemas with their sets of functional dependencies and their candidate keys.

3) Which of the functional dependencies in $\mathfrak{F}$ were lost in the decomposition?

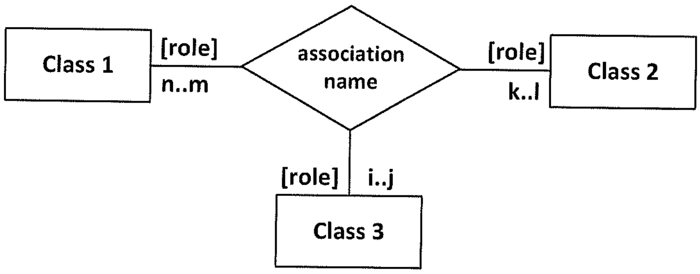
Appendix A: Notations for class diagrams

meta-notation:
[…] Optional element

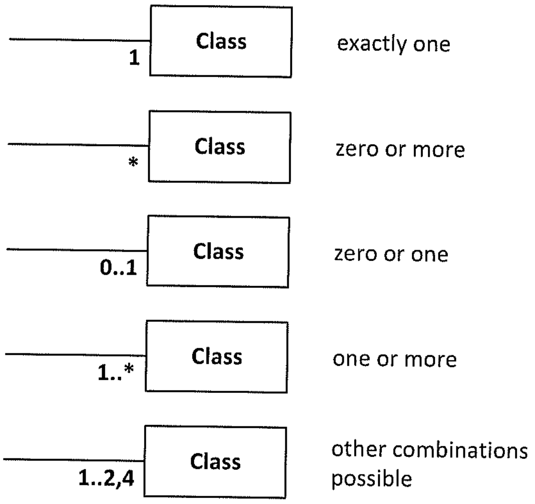
Class and Association



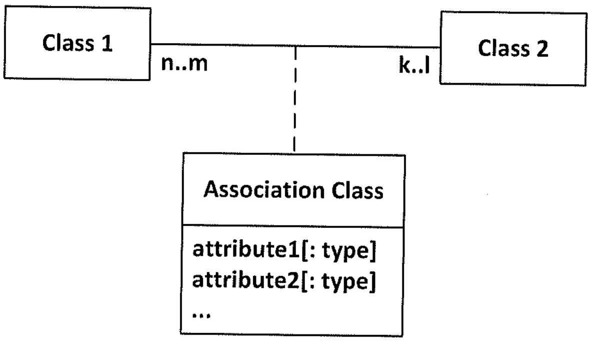
Ternary (or *n*-ary) association



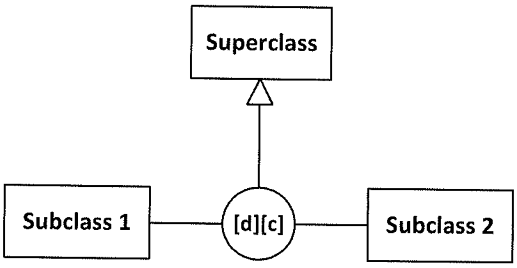
Multiplicity



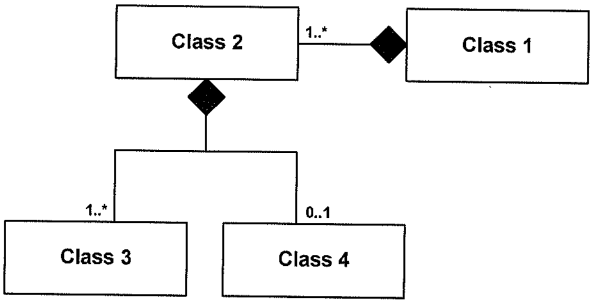
Association class



Generalization



Composition



Appendix B: Informal syntax for database schema

```
Informal syntax ('|' for choice and '[ ]' for optional):

Table (column [NOT NULL] [UNIQUE] [PK]
      [, column ... ]
      [, PK (column, ... ) ]
      [, FK (column, ...) REF table [ (column, ...) ]
          [, FK ...] ]
      [, CHECK (condition) ]
);

Examples of condition:
  column = value [ (OR | AND) [NOT] column <> value ] |
  column IS [NOT] NULL |
  column [NOT] IN (value, ...) |
  ...
```

Appendix C: Losless BCNF decomposition algorithm

Definition of BCNF:
A relational schema is in BCNF if for every nontrivial functional dependency the left-hand side is a superkey.

Decomposition algorithm:
Let R be a relational schema with a set of functional dependencies $\mathfrak{F}$.
Let $X \rightarrow Y$ be a functional dependency in $\mathfrak{F}$ which violates the BCNF constraint.

- Decompose R into
  $R_1(X^+)$
  $R_2(Z)$ with $Z = \{X\} \cup \{$attributes of R not in $X^+\}$.
- For $i = 1,2$:
  - determine $\mathfrak{F}_i$ for R_i by restricting $\mathfrak{F}^+$ to functional dependencies within R_i
  - if R_i is not in BCNF, recursively apply the algorithm